MULTISCALE UNCERTAINTY QUANTIFICATION FOR PHYSICS-BASED DATA-DRIVEN MATERIALS DESIGN AND OPTIMIZATION

A Dissertation
Presented to
The Academic Faculty

By

Anh V. Tran

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

December 2018

MULTISCALE UNCERTAINTY QUANTIFICATION FOR PHYSICS-BASED DATA-DRIVEN MATERIALS DESIGN AND OPTIMIZATION

Approved by:

Dr. Yan Wang, Advisor
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology

Dr. David McDowell
George W. Woodruff School of Mechanical Engineering, School of Materials Science and Engineering
Georgia Institute of Technology

Dr. Chaitanya Deo
George W. Woodruff School of Mechanical Engineering, School of Materials Science and Engineering
Georgia Institute of Technology

Dr. Hongyuan Zha
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Xin Sun
Energy and Transportation Science Division
Oak Ridge National Lab

Date Approved: November 9, 2018

The world will ask you who you are, and if you don't know, the world will tell you.

Carl Jung

Dedicated to Jing Sun and Alan Sun Tran.

ACKNOWLEDGEMENTS

I would like to acknowledge my wife, Jing Sun, who has been with me through many ups and downs since the time we have known each other, and for her kind encourage and sharing through difficult moments, and to Alan Sun Tran, who inspired me to strive endlessly.

I would like to express my sincere gratitude toward my academic advisor, Prof. Yan Wang for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. He has spent numerous hours to mentor and help me to improve the work.

I would like to thank the rest of my thesis committee: Prof. David McDowell, Prof. Chaitanya Deo, Prof. Hongyuan Zha, and Dr. Xin Sun, for their insightful comments, but also for the hard questions which encourage me to widen my research from various perspectives.

I am grateful to the generous financial support by the U.S. National Science Foundation under Grant No. CMMI-1306996, for without it, the research effort would not be possible.

I thank my fellow group members: Lijuan He, Dehao Liu, Justin Yoo, Yanglong Lu, Jesse Sestito, Michael Kempner, and other MSSE members for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last four years.

My sincere thanks also goes to Dr. John Furlan, Dr. Mohamed Garman, and Mr. Robert Visintainer at GIW Industries, who provided me an opportunity to join their team as engineering intern, and who gave access to the laboratory and research facilities.

Last but not the least, I would like to thank my family: my parents and to my sister for supporting me spiritually throughout writing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Uncertainty is a critical element in computational materials science. From the experimental perspective, the sources of uncertainty include measurement errors caused by instrument, operator, and sensing models, as well as the curve fitting in determining the parameters of constitutive material models. From the computational perspective, the choice of material models is a source of model-form uncertainty, whereas the parameters used in these constitutive material models are sources of parameter uncertainty. Both the model-form and parameter uncertainties can be considered as epistemic uncertainty, which is reducible as the knowledge about the material advances. On the other hand, irreducible or aleatory uncertainty originates from the random distribution of materials because of the nature of statistical thermodynamics. Thus, uncertainty quantification (UQ) is an important aspect in integrated computational materials engineering (ICME) tools for the credible predictions of physical quantities of interests. The major challenges of UQ in materials modeling are the curse of dimensionality and computational complexity.

In this research, efficient UQ methods are developed for both forward uncertainty propagation and prediction in ICME tools and the inverse process of materials design and optimization. A temporal-upscaling stochastic reduced-order model is developed to accelerate the uncertainty propagation beyond the time-scale limitation in simulations. The proposed method is demonstrated using kinetic Monte Carlo, molecular dynamics, and phase field simulations in the microstructural evolution problem. The stochastic collocation method is applied to study the effects of processing and thermodynamic parameters and efficiently quantify uncertainty in dendritic growth. It is demonstrated with the solidification process of Al-4wt%Cu using phase field model. For materials design and optimization under uncertainty, Bayesian optimization is extended to solve large-scale problems. A novel and accelerated Bayesian optimization method is proposed to tackle the mixed-integer optimization under known and unknown constraints, based on computationally expensive high-fidelity

simulations through the parallel usage of high-performance computing architecture. The proposed batch parallel Bayesian optimization method is demonstrated using several design optimization examples, including fractal metamaterials and structures.

CHAPTER 1
INTRODUCTION

Uncertainty is a critical element in computational materials science, and particularly in integrated computational materials engineering (ICME) tools. From the experimental perspective, the measurement error, as well as the curve fitting in determining parameters for constitutive material models, are sources of uncertainties. From the computational perspective, the choice of constitutive material models is a source for model-form uncertainty, whereas a source of parameter uncertainty is rooted from the experimental calibration. Both the model-form and parameter uncertainties described above can be considered as epistemic uncertainty, which is reducible as the knowledge about the material advances. On the other hand, there is another source of uncertainty in materials, which is irreducible and usually referred to as aleatory, originating from the spatial random nature of the material. The manifesto of aleatory uncertainty in ICME tools is the necessity of running ICME tools multiple times with different representative volume elements (RVEs), which is a realization of the stochastic microstructure. Thus, uncertainty quantification (UQ) is an important aspect in computational materials science, as ICME tools are typically used as a forward model to predict the quantities of interests (QoIs). As most of the usage of ICME tools is for time-dependent and dynamic problems, the UQ in ICME problems naturally becomes uncertainty propagation (UP) problems, in which the QoIs are random variables evolving over time.

While ICME tools are forward models because of their predictive capabilities, the materials design problem is the inverse problem, in which one seeks an optimal process that is corresponding to optimal material properties. Since uncertainty is an inevitable element of the ICME tools, the optimization in materials design problems must account for uncertainty during the optimization procedure. This leads to the optimization under

uncertainty in the context of computational materials science.

The pressing need for UQ, UP, and optimization under uncertainty in computational materials science is highly related to the well-known process-structure-property (PSP) chain. Because ICME tools are widely used to numerically establish the linkage between process-structure or structure-property, the uncertainty in the ICME tools must be accounted for in the inputs and must be quantified in the outputs, in order to successfully establish the linkage. Multiple sources of uncertainty are evolved during this process. The most apparent uncertainty source is the parameter uncertainty, in which one can vary different parameters to obtain different QoIs. For example, one can vary the processing parameters and observe the change in microstructure for the process-structure linkage. The second uncertainty source is the model-form uncertainty, in which different constitutive material models are used to simulate the QoIs. For instance, in molecular dynamic (MD) simulations, different atomic potential models can be used in the simulation systems. Obviously, different atomic potentials, which can be considered as the constitutive material models, yield different QoIs for the same simulation settings. While the model-form and parameter uncertainties are epistemic uncertainty, the third source of uncertainty is the inherent randomness of the microstructure, which is aleatory uncertainty. The material microstructure is non-deterministic, random, and can be described and represented statistically. The aleatory uncertainty of microstructure is mainly due to two reasons. First, the microstructure is usually not homogeneous. It means that different compositions may be obtained at different locations. Second, because of the limitation of processing techniques, distributions of microstructures are stochastic and exhibit natural variability. As a result, UQ is regarded as a useful and necessary tool to be used in concert with ICME tools.

However, UQ in ICME is a difficult problem to solve, because it carries challenges from both UQ and ICME. On one hand, the technical challenges in the UQ problems are the curse of high dimensionality and the complexity of the engineering simulations. If s is the number of sampling points per dimension, and d is the dimensionality of the prob-

lem, then s^d is the number of sampling points in the full tensor grid approach. The number of sampling points increases exponentially with respect to the dimension, creating the computational bottleneck for UQ. On the other hand, the ICME bottleneck roots in the complexity issues of ICME tools, including scarcity of data, input representation, noisy evaluation, and time-scale limit. The scarcity-of-data problem in computational materials science is due to two reasons. First, the ICME tools are computationally expensive. Second, there is a lack of experimental data to support the development of ICME tools, due to the time consuming experimental setup and significant effort to obtain the data. One can argue that with the rapid development of high-performance computing (HPC) systems, the computational cost to obtain the simulation data has been reduced drastically. However, the ICME tools are also getting more complex and take more time to obtain simulation results. The representation of ICME tool inputs also poses another challenge in studying the PSP linkage problems. For example, it is non-trivial to compare one process with another, in terms of material properties and performance. However, representing the processes, which sometimes include time series, thermal history, and different heat treatments, is a careful procedure. First, the redundancy must be avoided, which could lead to the curse of dimensionality. Second, the uniqueness of the representation must be preserved. The sole usage of continuous variables as the parameters in ICME tools is thus insufficient. This naturally leads to the incorporation of discrete and categorical variables in modeling the distinction between different processes. Thus, UQ problems in ICME context becomes more difficult. For example, the inverse design problems could potentially lead to a combinatorial or a mixed-integer problem, which could be NP-hard and becomes computationally intractable. The noisy ICME simulations also pose another challenge in the UQ context. Generally speaking, one can link the noise in the ICME tools with the noise in microstructures in the PSP linkage. The non-deterministic nature of ICME tools in modeling microstructures requires many simulations to be evaluated, which adds to the computational burden. Furthermore, material models for ICME are also subject to the complex multiscale behaviors

of materials. The hierarchical time and length scales need to be modeled simultaneously.

Both physics-based and data-driven paradigms can be utilized to solve the UQ problems in the context of computational materials science. On one hand, physics-based approaches typically rely on the ICME tools, after the verification and validation processes. UQ methods, such as Bayesian model average, have been used to capture model-form uncertainty. Other UQ methods, such as stochastic collocation, sparse grid, polynomial chaos expansion, and Gaussian process regression, can be applied to quantify parameter uncertainty of the ICME tools. Generally speaking, the parameter uncertainty problems are easier to solve than model-form uncertainty problems since more methods are available. Data-driven approaches, on the other hand, rely on the rich data resource to solve the UQ problems. The data can be obtained either through simulations or experiments. Because of the scarcity of experimental data, computational data are more accessible to researchers, particularly with the support of HPC systems. The significant advantage of data-driven approaches in UQ is the computational efficiency in prediction, in which the surrogate or metamodel can approximate the underlying ICME tools. Some well-known data-driven approaches, such as machine learning, are available to construct surrogate models. However, the major drawback of data-driven approaches is that the accuracy of the prediction is dependent on the size of the required data for surrogate model construction. High-dimensional problems need exponentially large datasets to train the surrogate models properly. One of the potential approaches to deal with the scarcity of data is to combine physics-based and data-driven methods. Physical knowledge can be injected into the formulation of surrogate models, as a "gray box" approach, to significantly reduce the training data size, as demonstrated in our previous work [1]. Hybrid approaches that aggregate both computational and experimental data through multi-fidelity modeling are also promising to mitigate the scarcity of data. The experimental data, which is typically scare and relatively more accurate, can serve as a high-fidelity resource, whereas the computational data can be regarded as a low-fidelity resource.

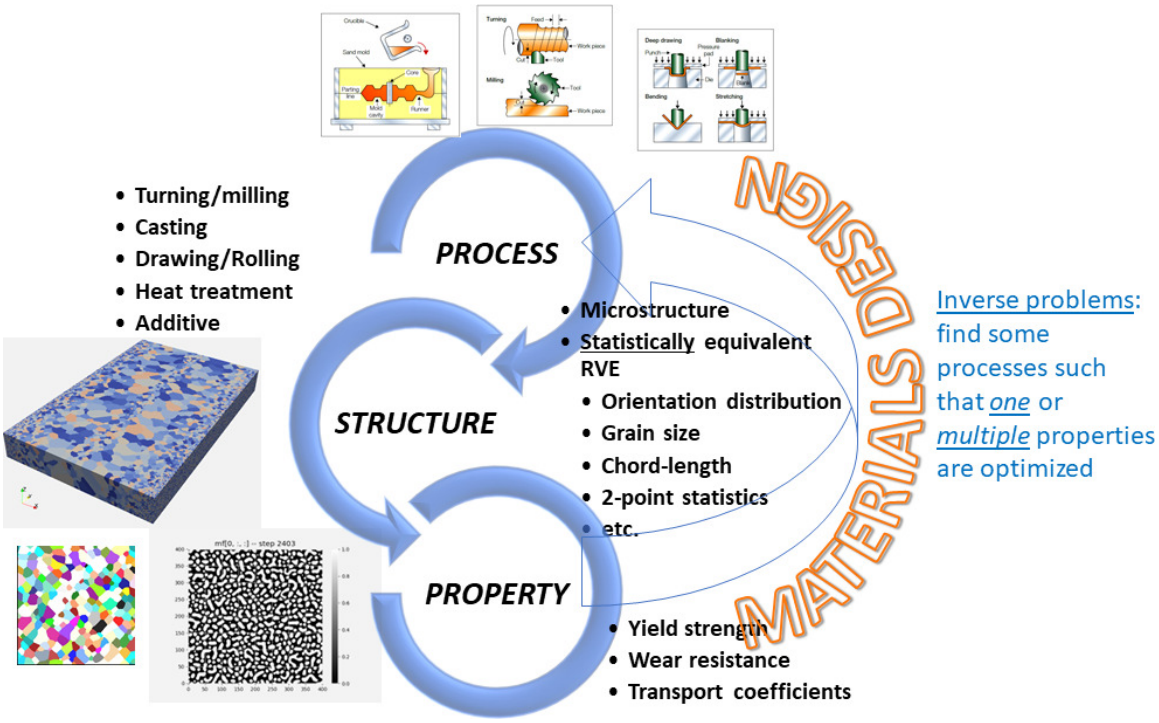


Figure 1.1: Overview of Process-Structure-Property (PSP) linkages.

Figure 1.1 shows an overview of the PSP linkage in materials science. A process can be any arbitrary manufacturing process, such as conventional casting, milling, drawing, or even additive manufacturing. It can be originated from the chemical composition space from the chemistry perspective, where the availability and quantity of elements in the periodic table can serve as the input parameters. This representation spans a vast collection of materials, including lightweight alloys, superalloys, high-entropy alloys, and shape-memory alloys. A process can be generally represented by a collection of categorical, discrete, and continuous process parameters. Each process would result in a collection of *statistically equivalent* microstructures, in which the statistical descriptors of the representative volume element (RVE) are similar. Many examples of microstructural descriptors are reviewed in [2, 3]. Some common microstructural descriptors include volume fraction, number of contiguous neighbor grains, cluster's nearest center distance, the orientation angle of a cluster's principal axis, total surface area, number of clusters, the equivalent radius, compactness, roundness, eccentricity, aspect ratio, rectangularity, and tortuosity. As

the microstructure sample varies upon the location of the specimen probe, a collection of statistical descriptors is necessary in describing the microstructure. The variation of these statistical descriptors constitutes the aleatory uncertainty in materials.

From the obtained microstructures, the structure-property linkage can be computationally established to predict the property of the microstructure with ICME tools, and eventually, bridge the gap from processing parameters to property. The materials property can be yield strength, density, wear resistance, transport coefficients, etc., depending on the application of the materials. In the materials design problem, one seeks to solve the inverse problem, in which the following question is posed: *What are the processes that correspond to one or multiple optimal materials properties?* The materials design question thus can be formulated as an optimization problem, which part of this dissertation is dedicated to solve.

There are four paradigms in establishing the PSP linkage of materials, as illustrated in Figure 1.2 [4]. It is noted that a later paradigm is dependent on the previous ones. The first paradigm is the experimental approach, where one seeks to map from process to property using experimental measurements. The experimental approach is regarded as the most reliable one, especially with today's advanced instruments. The second approach is the mathematical formulation of physics-based constitutive models that describe the phenomena. The third approach is a more recent advancement, which concerns with the computational implementations of ICME tools to build the PSP linkage. The computational model is often calibrated using experimental data, with constitutive models as kernels, thus is dependent on both experimental methods and theoretical constitutive models. The fourth paradigm is the most recent approach, which primarily concerns with using the data from experiments and computations to establish the PSP linkage in a fast and accurate manner. The fourth paradigm also opens up the possibility for the data-driven constitutive model, where dimensionality reduction techniques can be applied to model physics-based parameters as latent variables.

In this dissertation, we considered four problems in UQ, UP, and optimization under

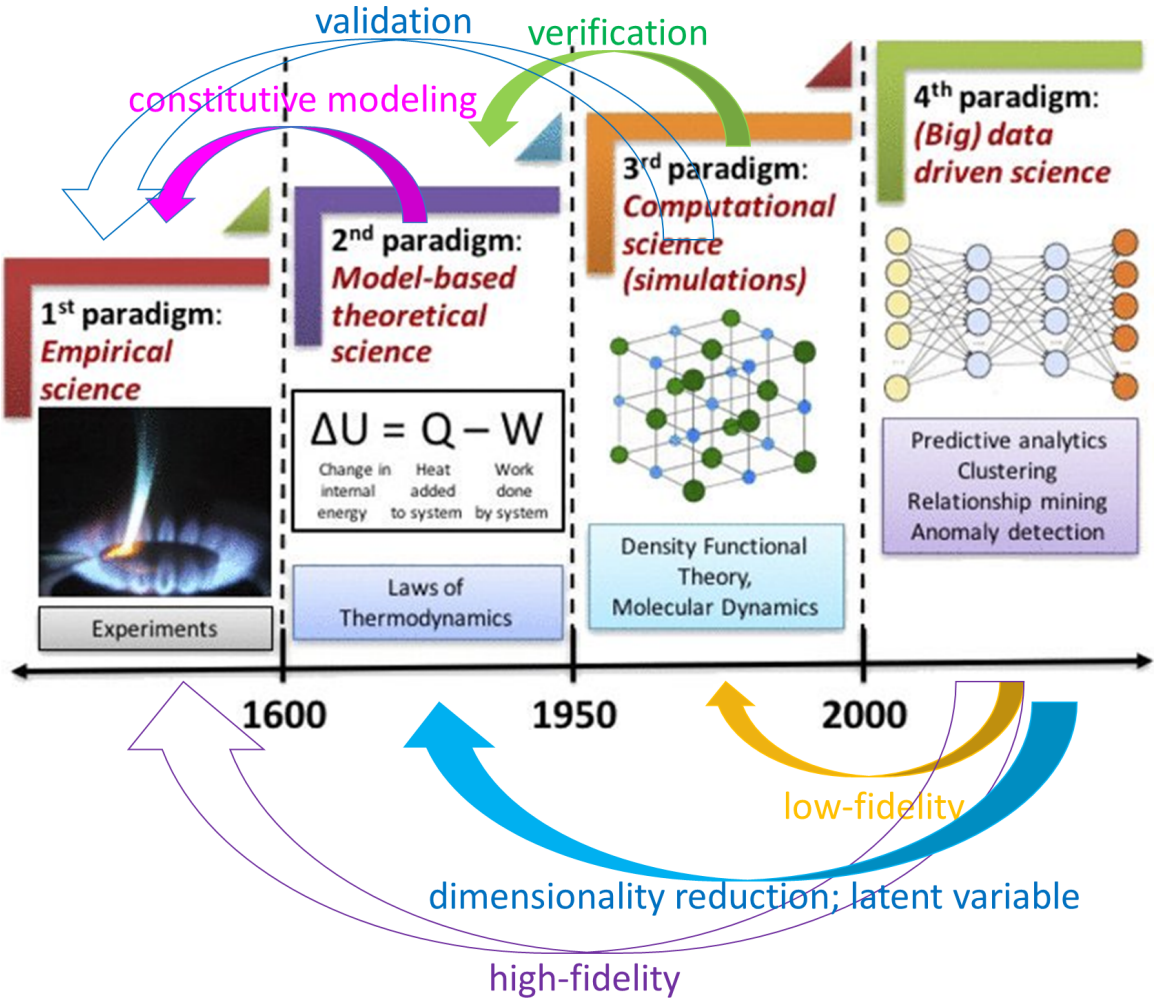


Figure 1.2: Four paradigms in establishing process-structure-property linkage: experiment, constitutive modeling, computational simulation, and machine learning approaches, with the later paradigms dependent on all previous ones.

uncertainty. The literature reviews are provided in each corresponding chapter with respect to the problem statement. In Chapter 2, we considered UP problems on fast time-scale and proposed a stochastic reduced-ordered model (ROM), which is constructed based on the underlying ICME tools and is capable of escaping the time-scale limit of the ICME tools. The proposed ROM is then demonstrated using phase field model (PFM), kinetic Monte Carlo (kMC) simulation, and molecular dynamics (MD) in modeling the microstructural evolution.

In Chapter 3, we applied the sparse grid (SG) and polynomial chaos expansion (PCE) methods to capture the variation of dendritic growth in the Al-4%Cu binary alloy solidification process. The dendrite growth is investigated using PFM as a function of processing and thermodynamic parameters, which correspond to epistemic and aleatory uncertainties, respectively.

In Chapter 4, we developed a fast Bayesian optimization (BO) method, which solves a general optimization problem and thus can be applied for any materials design problem. The method, called pBO-2GP-3B algorithm, aims to solve the optimization problem where the high-fidelity ICME tool is regarded as a functional evaluator in a HPC environment. The pBO-2GP-3BO algorithm is an optimization method based on the Gaussian process framework and quantifies uncertainty during the optimization process. It can significantly accelerate the optimization procedure using batch parallelization, which simulates concurrently a number of simulations, with know and unknown constraints. The novelties of the method are two-fold. First, the optimization problem, where high-fidelity ICME tools are used as the functional evaluator, can be efficiently solved through parallelism in the HPC environment, and multiple simulations are run in parallel. Second, the proposed method also handles unknown constraints, which must be evaluated through simulations. The proposed pBO-2GP-3BO algorithm is demonstrated using a few synthetic problems and a high-fidelity computational fluid dynamics (CFD) simulation to predict wear in centrifugal slurry pumps.

In Chapter 5, we developed a mixed-integer BO method to solve for the mixed-integer constrained optimization problem. The method aims to solve a general constrained optimization problem, in which the constraints are known beforehand, and discrete variables are involved. The proposed method is demonstrated with using several synthetic examples, as well as two engineering applications in designing mechanical and auxetic metamaterials using finite element analysis (FEA).

The technical contribution for the stochastic ROM is a novel ROM formulation that employs data-driven approach to leverage the time-scale limit of the ICME tools. The stochastic ROM is successfully demonstrated using PFM, kMC, and MD simulations.

The technical contribution of the large-scale UQ study on PFM is the revealing of the impact of processing and thermodynamic parameters on the dendrite growth and morphology.

The technical contributions in the optimization chapters are two novel BO methods. In the mixed-integer BO method, we propose a novel BO framework that is based on categorical regression by decomposing a large dataset according to an ordered collection of discrete variable. Then, a Gaussian mixture model is formulated to estimate a posterior means and variances in the classical GP framework. The proposed mixed-integer BO method clearly shows a higher performance in optimization problems than genetic algorithms. In the batch-parallel constrained BO method, we propose another novel BO framework that efficiently uses the HPC architecture to solve optimization problems in a parallel manner. Furthermore, this BO framework also handles unpredictable crash or ill-conditioned forward models. The problem is formulated as blind or unknown constraints.

CHAPTER 2

STOCHASTIC REDUCED-ORDER MODEL FOR TEMPORAL UPSCALING OF UNCERTAINTY PROPAGATION IN MATERIALS MODELING

Integrated Computational Materials Engineering (ICME) models and simulations are among the most useful toolboxes to develop and study materials. However, time-scale issue and uncertainty quantification remain two critical problems in applying ICME models at the multi-scale level. In this chapter, we propose a time-upscaling stochastic reduced-order model (ROM) to propagate the uncertainty based on the stochastic process of 1D non-linear Langevin equation with Gaussian noise, which describes the evolution of the probability distribution function based on the Fokker-Planck equation. The proposed method is demonstrated with three different ICME models, including kinetic Monte Carlo, phase-field, and molecular dynamics simulations. Once the drift and diffusion coefficients are calibrated, the evolutions of the statistical microstructural descriptors are accurately predicted using the ROM.

2.1 Introduction

Simulating the dynamic behaviors of material systems is one of the most important tasks for materials modeling. Quantities of interests (QoIs) are mostly related to the evolution of the systems along time. For instance, at atomistic scale, thermodynamic and mechanical properties of materials can be predicted by molecular dynamics (MD) simulation. Properties related to transport and phase transformation can be estimated from kinetic Monte Carlo (kMC) simulation. At mesoscale, solidification processes are simulated by phase field (PF) or cellular automaton models. The evolution of dislocations in crystals can be simulated with discrete or continuous dislocation dynamics models. Given the model-form and parameter uncertainty associated with these models, the credibility of simulation pre-

dictions largely relies on how the uncertainty can be effectively quantified. The major challenges of uncertainty quantification (UQ) for materials modeling are associated with the high dimensionality of models and complexity of uncertainty propagation during dynamics simulations.

Here, the uncertainty propagation problem is considered, where uncertainty associated with input parameters and initial conditions of material systems evolves along time during the dynamics simulation. Direct modeling of uncertainty propagation in material systems suffers from the time-scale issue, because a very short time step (could be as short as femtoseconds) is usually needed for each iteration in order to obtain the required fidelities of material systems and hundreds of thousands of iterations are typical. Regular UQ methods could add significant overhead costs to these simulations, which themselves already are computationally expensive to obtain meaningful QoIs.

Several methods have been proposed to propagate uncertainty on a fast time-scale, including stochastic ROM, Kramers-Moyal expansion and Fokker-Planck equation, and parallel-in-time methods. The stochastic ROM methods aim to approximate the models with statistically equivalent QoIs, where the computational cost is significantly reduced compared to the physics-based models. The Kramer-Moyal expansion and Fokker-Planck equation model the uncertainty propagation using a partial differential equation with second- or higher-order derivative, where the QoIs' probability density functions (PDFs) evolve gradually. The parallel-in-time methods pose another parallelism for the numerical solvers, in which discretization in time-domain can be solved simultaneously, thus mitigating the impact of the time-scale issue. The literature will be further discussed in Section 5.2.

To alleviate the time-scale issue in propagating uncertainty in materials modeling, in this chapter, a time-upscaling stochastic ROM is introduced using a stochastic differential equation to propagate the PDFs of QoIs. The advantages of the proposed ROM is two-fold. First, the uncertainty of QoI is quantified and propagated based on the forward Fokker-

Planck equation, which is a partial differential equation, with calibrated parameters. Second, since the time scale of the Fokker-Planck equation is independent of that in material models, the numerical solution of the Fokker-Planck equation can be obtained much faster than the direct simulations of materials. The QoIs can be any microstructural descriptor of the materials, parameters of the constitutive models, or constraints in the calibration process.

In this chapter, we approach the time-scale problem for uncertainty propagation by introducing a stochastic ROM for the QoIs with time upscaling. The parameters of the stochastic ROM are trained either analytically or numerically. The evolution of a QoI as time series is divided into two parts. The first part is used to train and calibrate the ROM, whereas the second part is used to test the ROM performance. The underlying assumption is that the QoIs can be modeled as stochastic processes, where PDFs of QoIs can evolve with respect to time. The significant difference between the ROM and the direct propagation approach is their computational cost, where the former is much lower.

The outline of the chapter is as follows. Section 5.2 provides a literature review of related work. The mathematical foundation of the proposed methodology and the numerical procedure in applying the proposed method is described in Section 2.3. Section 2.4 provides three examples of kMC, PF, and MD simulations. The advantage and possible improvements of the proposed methods are discussed in Section 3.6. Section 2.6 concludes the chapter.

2.2 Related work

In this section, literature review on uncertainty propagation and time-upscaling methods is provided to highlight the state-of-the-art techniques in assessing the evolution of QoI using UQ methods.

2.2.1 Stochastic reduced-order models

Grigoriu [5] constructed a stochastic ROM with simple random functions to approximate an arbitrary random functions, where statistical discrepancy are minimized. Sarkar et al. [6] applied the method of Grigoriu [5] to quantify uncertainty in a corroding system and compare against sampling-based approaches. Mignolet and Soize [7] proposed another stochastic ROM for both model and parameter uncertainty in a stochastic finite element approach. This nonparametric approach accounts for both model and parameter uncertainty, compared to only parameter uncertainty in Ghanem and Spanos [8].

2.2.2 Kramers-Moyal expansion and Fokker-Planck equation

The partial differential equation of PDFs based on the Kramers-Moyal expansion models the evolution of probability distributions. Fokker-Planck equation is a special case of it with only the first- and second-order spatial derivatives considered. The calibration of the associated drift and diffusion coefficients from observation is an important topic. Numerical estimations of the Fokker-Planck drift and diffusion coefficients with high and low sampling rates for time-series data have been studied extensively in the literature (for example, Pienke et al. [9] [10] [11] [12], Sura and Barsugli [13], Ragwitz et al. [14] [14]), where the data with low sampling rates are insufficient to estimate the coefficients accurately. Our proposed method is able to bypass the technical challenges of the sampling rates, since it solves the inverse problem, in which the coefficients are first parameterized, then optimized to minimize the difference between the simulated and calibrated PDFs.

2.2.3 Parallel-in-time methods

Time parallelism methods have been introduced to accelerate dynamics simulations [15], including multiple shootings, domain decomposition and waveform relaxation, multigrid, and direct time parallel methods. Nievergelt [16] proposed the first shooting type approach by pure time decomposition. Chartier and Phillipe [17], Saha et al. [18], May-

day and Turicini [19], and Guillaume [20] further developed and analyzed the method, often referred to as parareal algorithm in the literature. The iterative domain decomposition in space-time domains also have received attentions from many researchers, including Lelarasmee [21], Gander et al. [22] [23]. Multigrid methods is another iterative approach that is not naturally parallel, but their components can be parallelized in the entire space-time domain [16]. Notable work includes Hackbusch [24], Lubich and Ostermann [25], Horton and Vandewalle [26], Emmett and Minion [27], Neumüller [28]. Direct solvers are the last in the four classes of time parallelism methods. Prior work incudes Miranker and Liniger [29], Axelson and Verwer [30], Womble [31], Worley [32], Sheen et al. [33], Maday and Ronquist [34], Christlieb et al. [35], Güttel [36]. Despite their success, the parallel-in-time methods described above would only scale if the computational resources are sufficient. Consequently, time acceleration is achieved only if the high-performance computing infrastructure is available.

2.3 Methodology

In this section, the proposed stochastic ROM method with time upscaling is introduced, the Kramers-Moyal expansion and the Fokker-Planck equation are introduced, along with the mathematical formulation, parametrization, calibration, and forward propagation using ROM.

2.3.1 Kramers-Moyal expansion and Fokker-Planck equation

The Kramers-Moyal expansion for the PDF $f(x,t)$ associated with random variable X can be written as [37]

$$\frac{\partial f(x,t)}{\partial t} = \sum_{j=1}^{n} \left(\frac{-\partial}{\partial x}\right)^{j} D^{(j)}(x,t) f(x,t). \tag{2.1}$$

where n is the number of truncated terms in Kramers-Moyal expansion, $D^{(j)}(x,t)$ is the Kramers-Moyal expansion coefficient. For the non-Markovian processes, the coefficients

$D^{(j)}$ depend on the values of the stochastic variable $X(t')$ at all earlier times for $t' < t$. On the contrary, the coefficients do not depend on the earlier time in Markovian processes.

Theorem 1 (Pawula's theorem[37])**.** *The Kramers-Moyal expansion may stop either after the first term or after the second term; if it does not stop after the second term, then it must contain an infinite number of terms.*

With only the first two terms, the Kramers-Moyal expansion is reduced to the Fokker-Planck equation as

$$\frac{\partial f(x,t)}{\partial t} = -\frac{\partial}{\partial x}\left[D^{(1)}(x,t)f(x,t)\right] + \frac{\partial^2}{\partial x^2}\left[D^{(2)}(x,t)f(x,t)\right], \tag{2.2}$$

where $D^{(1)}$ and $D^{(2)}$ are the drift and diffusion coefficients, respectively, and both are spatio-temporal functions.

Suppose the evolution of the random variable X, which is the QoI, can be modeled by the Fokker-Planck equation. The evolution of mean and variance can be modeled through the two following theorems.

Theorem 2. *The expectation of X, denoted as $\mathbb{E}[X]$,*

$$\mathbb{E}[X] := \int_{-\infty}^{\infty} x f(x,t) dx, \tag{2.3}$$

must satisfy

$$\frac{\partial}{\partial t}\mathbb{E}[X] = \int_{-\infty}^{\infty} D^{(1)}(x,t) f(x,t) dx. \tag{2.4}$$

If the drift coefficient is a temporal function, only i.e. $D^{(1)}(x,t) = D^{(1)}(t)$, then

$$\frac{\partial}{\partial t}\mathbb{E}[X] = D^{(1)}(t). \tag{2.5}$$

Proof. Assume vanishing boundary conditions at an exponential rate of the PDF, i.e. $f(x,t) \propto e^{-x^2} \Rightarrow \lim_{x\to\pm\infty} f(x,t) = 0$. Here, we integrate by part and utilize the vanishing bound-

ary conditions of the PDF $f(x,t) \to 0$ as $x \to \pm\infty$.

$$\begin{aligned}
\tfrac{\partial}{\partial t}\mathbb{E}[X] &= \tfrac{\partial}{\partial t}\int_{-\infty}^{\infty} x f(x,t)dx = \int_{-\infty}^{\infty} x\tfrac{\partial f}{\partial t}dx \\
&= \int_{-\infty}^{\infty}\left[-x\tfrac{\partial}{\partial x}\left(D^{(1)}f\right) + x\tfrac{\partial^2}{\partial x^2}\left(D^{(2)}f\right)\right]dx \\
&= -\int_{-\infty}^{\infty} x\tfrac{\partial}{\partial x}\left(D^{(1)}f\right)dx + \int_{-\infty}^{\infty} x\tfrac{\partial^2}{\partial x^2}\left(D^{(2)}f\right)dx \\
&= -[xD^{(1)}f]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} D^{(1)}f dx + [x\tfrac{\partial}{\partial x}(D^{(2)}f)]_{-\infty}^{\infty} - \int_{-\infty}^{\infty}\tfrac{\partial}{\partial x}(D^{(2)}f)dx \\
&= -[xD^{(1)}f]_{-\infty}^{\infty} + \int_{-\infty}^{\infty} D^{(1)}f dx + [x\tfrac{\partial}{\partial x}(D^{(2)}f)]_{-\infty}^{\infty} - [(D^{(2)}f]_{-\infty}^{\infty} \\
&= \int_{-\infty}^{\infty} D^{(1)}(x,t)f(x,t)dx
\end{aligned} \tag{2.6}$$

If the drift coefficient is a temporal function $D^{(1)}(x,t) = D^{(1)}(t)$, then the last equation simplifies to

$$\frac{\partial}{\partial t}\mathbb{E}[X] = D^{(1)}(t), \tag{2.7}$$

as $\int_{-\infty}^{\infty} f(x,t)dx = 1$. □

Theorem 3. *Assume that the drift coefficient is a temporal function, i.e.* $D^{(1)}(x,t) = D^{(1)}(t)$*, then the variance of* X*, denoted as* $\mathrm{Var}[X]$*,*

$$\mathrm{Var}[X] := \int_{-\infty}^{\infty}(x - \mathbb{E}[X])^2 f(x,t)dx, \tag{2.8}$$

must satisfy

$$\frac{\partial}{\partial t}\mathrm{Var}[X] = 2\int_{-\infty}^{\infty} D^{(2)}(x,t)f(x,t)dx \tag{2.9}$$

If the diffusion coefficient is also a temporal function, i.e. $D^{(2)}(x,t) = D^{(2)}(t)$*, then*

$$\frac{\partial}{\partial t}\mathrm{Var}[X] = 2D^{(2)}(t). \tag{2.10}$$

Proof. First, consider

$$\begin{aligned}
\frac{\partial}{\partial t}\mathbb{E}[X^2] &= \frac{\partial}{\partial t}\int_{-\infty}^{\infty} x^2 f(x,t)dx = \int_{-\infty}^{\infty} x^2 \frac{\partial f}{\partial t}dx = \int_{-\infty}^{\infty} x^2 \left[-\frac{\partial}{\partial x}\left(D^{(1)}f\right) + \frac{\partial^2}{\partial x^2}\left(D^{(2)}f\right)\right]dx \\
&= \int_{-\infty}^{\infty} x^2 \left[-\frac{\partial}{\partial x}\left(D^{(1)}f\right) + \frac{\partial^2}{\partial x^2}\left(D^{(2)}f\right)\right]dx \\
&= \int_{-\infty}^{\infty} -x^2 \frac{\partial}{\partial x}\left(D^{(1)}f\right)dx + \int_{-\infty}^{\infty} x^2 \frac{\partial^2}{\partial x^2}\left(D^{(2)}f\right)dx \\
&= -[x^2 D^{(1)}f]_{-\infty}^{\infty} + 2\int_{-\infty}^{\infty} xD^{(1)}f dx + [x^2 \frac{\partial}{\partial x}(D^{(2)}f)]_{-\infty}^{\infty} - 2\int_{-\infty}^{\infty} x\frac{\partial}{\partial x}(D^{(2)}f)dx \\
&= -[x^2 D^{(1)}f]_{-\infty}^{\infty} + 2\int_{-\infty}^{\infty} xD^{(1)}f dx \\
&\quad +[x^2 \frac{\partial}{\partial x}(D^{(2)}f)]_{-\infty}^{\infty} - 2[xD^{(2)}f]_{-\infty}^{\infty} + 2\int_{-\infty}^{\infty} D^{(2)}f dx \\
x &= 2\int_{-\infty}^{\infty} xD^{(1)}f dx + 2\int_{-\infty}^{\infty} D^{(2)}(x,t)f(x,t)dx
\end{aligned} \tag{2.11}$$

Observe that if the drift coefficient is a temporal function $D^{(1)}(x,t) = D^{(1)}(t)$, then by Theorem 2, $\frac{\partial}{\partial t}\mathbb{E}[X] = D^{(1)}(t)$. Thus,

$$\begin{aligned}
2\int_{-\infty}^{\infty} xD^{(1)}f dx &= 2\int_{-\infty}^{\infty} xD^{(1)}(t)f(x,t)dx = 2D^{(1)}(t)\int_{-\infty}^{\infty} xf(x,t)dx \\
&= 2D^{(1)}(t)\int_{-\infty}^{\infty} xf(x,t)dx = 2D^{(1)}(t)\mathbb{E}[X] \\
&= 2\frac{\partial \mathbb{E}[X]}{\partial t}\mathbb{E}[X].
\end{aligned} \tag{2.12}$$

If the diffusion coefficient is also a temporal function $D^{(2)}(x,t) = D^{(2)}(t)$, then Equation 2.11 becomes

$$\frac{\partial}{\partial t}\mathbb{E}[X^2] = 2\frac{\partial \mathbb{E}[X]}{\partial t}\mathbb{E}[X] + 2D^{(2)}(t). \tag{2.13}$$

After a few algebraic manipulation, we obtain

$$\frac{\partial}{\partial t}\mathrm{Var}[X] = \frac{\partial}{\partial t}(\mathbb{E}[X^2] - [\mathbb{E}[X]]^2) = \frac{\partial}{\partial t}\mathbb{E}[X^2] - 2\frac{\partial \mathbb{E}[X]}{\partial t}\mathbb{E}[X] = 2D^{(2)}(t). \tag{2.14}$$

□

Theorem 4 ([38], [11], [12])**.** *Denote the n^{th} central moment as $M^{(n)}(x,t)$, then*

$$\frac{\partial}{\partial t}M^{(n)}(x,t) = n!D^{(n)}(x,t) \tag{2.15}$$

It is noted that Theorem 4 follows directly from the Taylor series expansion in deriving Kramers-Moyal expansion.

The one-dimensional (1D) non-linear Langevin equation for stochastic variable X, which is the QoI in this chapter, is equivalent to the Fokker-Planck equation. The 1D non-linear Langevin equation reads [37]

$$\dot{X} = h(X,t) + g(X,t)\Gamma(t), \tag{2.16}$$

where the Langevin force $\Gamma(t)$ is assumed to be a Gaussian random variable with zero mean and δ correlation function, i.e.

$$\langle \Gamma(t) \rangle = 0, \quad \langle \Gamma(t)\Gamma(t') \rangle = \delta(t-t'). \tag{2.17}$$

There are two alternative ways to interprete the drift and diffusion coefficients of the Fokker-Planck equation. The first is based on Itô calculus and the second is based on Stratonovich calculus, depending on the existence of spurious or noise-induced drift [37]. For Itô calculus, $D^{(1)}(x,t) = h(x,t)$, whereas for Stratonovich calculus, $D^{(1)}(x,t) = h(x,t) + \frac{\partial g(x,t)}{\partial x} D^{(2)}(x,t)$. For both Itô and Stratonovich calculus, $D^{(2)}(x,t) = g^2(x,t)$. In the scope of this chapter, Stratonovich calculus is used to interpret the stochastic process of QoI.

The PDF for a QoI is numerically propagated along time, using Fokker-Planck equation with calibrated coefficients and initial conditions. There are two important elements in constructing the stochastic ROM.. First, the Fokker-Planck equation coefficients must be trained. Second, the initial conditions must be constructed numerically. The forward Fokker-Planck equation then can be solved using the calibrated coefficients and the initial conditions, and thus the QoI can be propagated. During the training, the initial PDFs and evolution of PDFs associated with the QoIs are obtained by running the original materials simulation models. The Fokker-Planck equation coefficients are calibrated to minimize the

Kullbeck-Leibler divergence between the PDFs predicted by the Fokker-Planck equations and the PDFs obtained from direct simulations. After training, the ROMs can be used to predict the PDFs of QoIs for longer periods of time, independently from the original material models.

2.3.2 Training Fokker-Planck coefficients

The numerical solution of Fokker-Planck equation can be solved by mainly two ways: finite element method and finite difference method. Both of them suffer the curse of dimensionality in high-dimensional space. However, in the scope of this chapter, only 1D stochastic process is concerned. This section outlines the numerical procedure to propagate the uncertainty in PDF of the QoI using ROM.

The training of drift, diffusion, and high-order coefficients for the time-upscaling stochastic ROM can be carried out in at least two ways: analytical and numerical. Based on Theorem 4 to estimate the coefficients. However, in practice, direct application of Theorem 4 faces challenges from both spatio and temporal dimension. First, the number of QoI observations is often not sufficient in practice to approximate the central moments well enough along the spatial dimension. Second, the sampling time is often sparse along time dimension. As a result, numerical estimations based on derivatives to approximate coefficients, based on Theorem 4, are often noisy and oscillatory, creating numerical challenges to construct the ROM model.

The first approach to train coefficients is based on Theorems 2 and 3 analytically. Observe that the QoI can be noisy in both temporal and spatial dimensions. One approach to reduce the effect of noise is to exclude the spatial variable x in the coefficients and simplify the coefficients to temporal function t, i.e. $D^{(n)}(x,t) = D^{(n)}(t)$. Excluding the spatial variable implies that the drift and diffusion coefficients are constant throughout the modeled spatial domain. The assumption of temporal-function coefficients is supported from the physics-based perspective. Materials distribution can be stable if a clear trend with respect

to time is observed and can be anticipated in the future.

This approach has the advantages of robustness subject to noisy QoI observation, because the central moment is calculated as a time-dependent quantity. Thus the numerical approximations converge to the central moment in probability by the weak law of large number [39].

The analytical approach built on Theorem 2 and Theorem 3 states that if the first two central moments are well constructed, then the drift and diffusion can be approximated by linear regression of mean and variance with respect to time, respectively.

The second approach to train the coefficients is based on the minimization of a loss function, after the coefficients are parameterized. The loss function can be defined with respect to the training PDF from direct simulation and the predicted PDF using Fokker-Planck equation with parameterized coefficients, for example, polynomial approximations, and minimize the loss function using an optimization algorithm. Compared to the analytical approach described previously, this numerical approach allows more flexible approximation forms of the coefficients.

The loss functions can be described as a distance between the training and predicted PDFs at a fixed time step, or at multiple time steps, where the predicted PDF with parameterized coefficients is compared with the training PDF. Mathematically, the loss functions can be expressed as either

$$l_1(x) = d\left(p^{(\tau)}_{\text{training}}(x), p^{(\tau)}_{\text{predicted}}(x)\right), \text{ at } t = \tau \text{ for a fixed time step } \tau, \tag{2.18}$$

or

$$l_2(x) = \sum_{j=1}^{n} w_j d\left(p^{(\tau_j)}_{\text{training}}(x), p^{(\tau_j)}_{\text{predicted}}(x)\right), \text{ at } t = \tau_j \text{ for multiple time steps } (\tau_j)_{j=1}^{n} \tag{2.19}$$

where $d(\cdot,\cdot)$ is the distance between two PDFs. Typical distances $d(\cdot,\cdot)$ could be mathematical distance, or statistical distance, such as l_p, L^P norms, Kullbeck-Leibler divergence,

Wasserstein distance, etc.

2.3.3 Regularization for initial condition

Regularizing the empirical initial PDF is a necessary step to solve the forward Fokker-Planck equation with parameterized coefficients. Without regularization, the forward solution of Fokker-Planck equation quickly diverges after a few steps.

Here the ridge regression method, which falls under the class of Tikhonov regularization, is applied to smoothen out the empirical initial PDF for the forward Fokker-Planck equation. The Tikhonov regularization method is discussed in details from Sticket [40].

The goal of regularization is to seek for an approximated PDF $\hat{f}(x)$ on a bounded domain $x \in \mathbb{R}$ that minimizes the penalized least squares function as

$$Q(\hat{f}) = \int_{x_1}^{x_N} \left| \frac{\hat{f}(x) - f(x)}{f(x)} \right|^2 dx + \lambda \int_{x_1}^{x_N} \left(\frac{\partial^d \hat{f}(x)}{\partial x^d} \right)^2, \tag{2.20}$$

where λ is the regularization parameter, and d is the order of derivative.

Using the linear algebra formalism, the objective function Q can be expressed as

$$Q = (\mathbf{M}\hat{\mathbf{f}} - \mathbf{f})^{\mathbf{T}}\mathbf{F}^{-\mathbf{2}}\mathbf{B}(\mathbf{M}\hat{\mathbf{f}} - \mathbf{f}) + \lambda(\mathbf{E}\hat{\mathbf{f}})^{\mathbf{T}}\tilde{\mathbf{B}}(\mathbf{E}\hat{\mathbf{f}}), \tag{2.21}$$

where $\boldsymbol{E}$ is the derivative matrix of any order, $\mathbf{M}$ is the mapping matrix, $\mathbf{B}$ is the midpoint rule integration matrix, $\tilde{\mathbf{B}}$ is a subset of $\mathbf{B}$, and $\mathbf{F} = \text{diag}(\mathbf{f})$ is the observation matrix in diagonal form.

Setting $\mathbf{M} = \mathbf{F}^{-\mathbf{2}}\mathbf{B} = \tilde{\mathbf{B}} = \mathbf{I}$ and solving $\frac{\partial Q}{\partial f} = 0$, we obtain the simplest form of smoothing by regularization, as

$$\hat{\mathbf{f}} = (\mathbf{I} + \lambda \mathbf{E}^{\mathbf{T}}\mathbf{E})^{-\mathbf{1}}\mathbf{f}. \tag{2.22}$$

2.3.4 Finite difference Fokker-Planck solver

The numerical solutions of Fokker-Planck equation can be obtained by mainly two methods: finite element and finite difference. Here, the finite difference scheme is applied. The training PDFs can be obtained from material simulation models in several ways, such as polynomial chaos expansion, stochastic collection, Latin hypercube sampling, and Monte Carlo sampling.

To construct the initial condition of the forward Fokker-Planck equation, attention must be paid in two aspects. First, the initial PDF is constructed at a time step $t = \tau_0$, where the stable trend can be detected in the QoI behavior. In materials modeling, the prediction of QoIs is typically unreliable during the first stage of simulations. The length of this unpredictable stage can be estimated through the analysis of the central moments of QoIs as time-dependent quantities. To avoid the unstable initial condition, the initial PDF is constructed after the trend for QoI is clear. Second, the initial PDF must be smoothened by applying a Tikhonov regularization method. Otherwise, the noise would quickly lead to a divergent solution of Fokker-Planck equation.

The forward Fokker-Planck equation is described as Equation 2.2, where the initial condition is $\boldsymbol{f}(x, t = \tau_0)$. The numerical implementation is developed based on the algorithm and computer code of [41], which calculates derivatives of any degree with any arbitrary order of accuracy over a uniform grid. Rewritting Equation 2.2 in a linear algebra formalism, we have

$$\dot{\boldsymbol{f}}(x,t) = -\boldsymbol{E}^{(1)}\left[D^{(1)}(x,t)\boldsymbol{f}(x,t)\right] + \boldsymbol{E}^{(2)}\left[D^{(2)}(x,t)\boldsymbol{f}(x,t)\right], \tag{2.23}$$

Discretizing in time dimension, Equation 2.23 can be numerically solved explicitly using the Runge-Kutta method, or implicitly using Crank-Nicolson method [42].

2.4 Applications and demonstrations

In this section, the proposed ROM is demonstrated using three examples: kMC simulation in Section 2.4.1, PF simulation in Section 2.4.2, and MD simulation in Section 2.4.3. In the kMC example, the selected QoI is the grain area. In the PF example, the selected QoI is the chord-length. Both are used as microstructural descriptors to describe the microstructure. In the MD example, the selected QoIs are the total mean-displacements and enthalpy of the simulation cell.

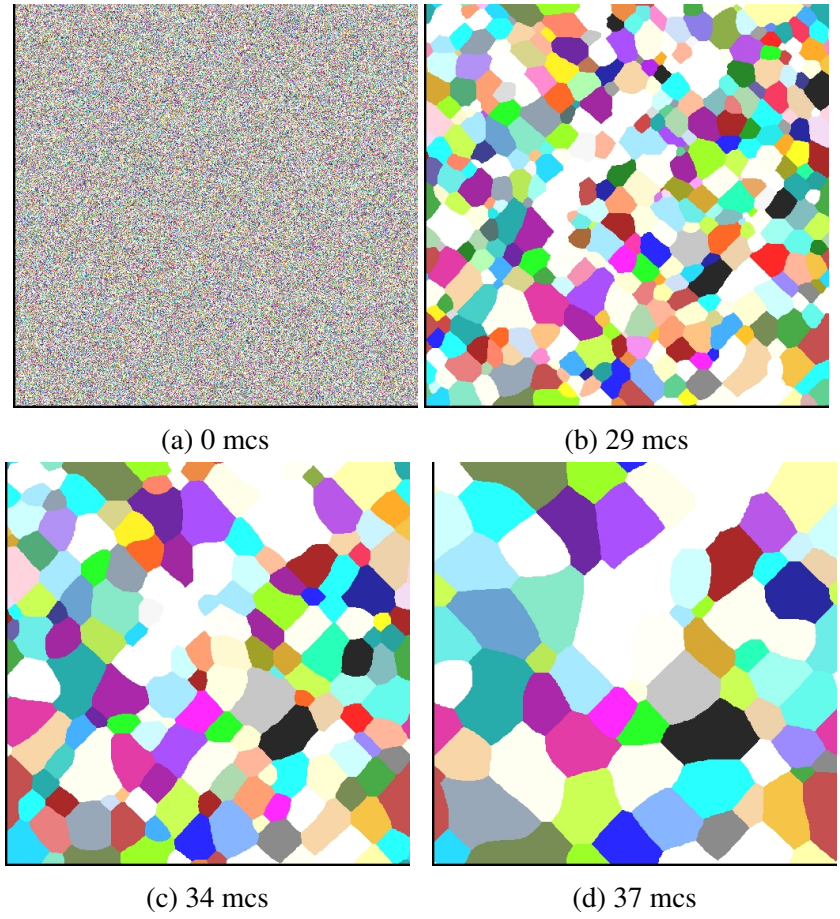
2.4.1 Kinetic Monte Carlo simulation: hybrid Potts-phase field simulation for grain growth

In this example, the hybrid Potts - phase field model from Homer et al. [43] based on kinetic Monte Carlo SPPARKS framework [44] is used to investigate the evolution of the grain area during the grain growth. The drift and diffusion coefficients are calibrated using Theorems 2 and 3. The initial and training PDFs are constructed using the kernel density estimation method with the normal kernel distribution. The selected bandwidth is optimal for the normal kernel density [45]. The initial PDF is constructed and regularized using the Tikhinov regularization as described in Section 2.3.3 to reduce the probability of divergent Fokker-Planck solution.

Figure 2.1 shows the evolution of microstructure using kMC. Figure 2.2 shows the evolution of the QoI using Fokker-Planck equation, where the QoI is grain area based on the kMC simulation results. The calibrated density is the last PDF used to train the stochastic ROM. The final density denotes the last PDF, obtained from direct simulations, to evaluate the performance of the trained stochastic ROM. The final density is not used for training the stochastic ROM.

In Figure 2.2a, the initial PDF of grain area is peaked at the size of approximately 5000 pixel^2, corresponding to Figure 2.1a, as the grains are fairly small and uniform. This is shown by the small variance of the QoI in Figure 2.1a. As the simulation continues, the

grain grows larger, and the variance of the QoI PDF increases accordingly. Figure 2.1d and Figure 2.2d present the microstructure and its corresponding QoI's PDF after the Fokker-Planck coefficients have been calibrated, respectively. In Figure 2.2d, the testing PDF and evolved Fokker-Planck PDF after calibration agree very well at the later testing time. This demonstrates if the Fokker-Planck coefficients are well-trained, a prediction about the evolution of the microstructural descriptor using the trained ROM can be made with a good level of accuracy.



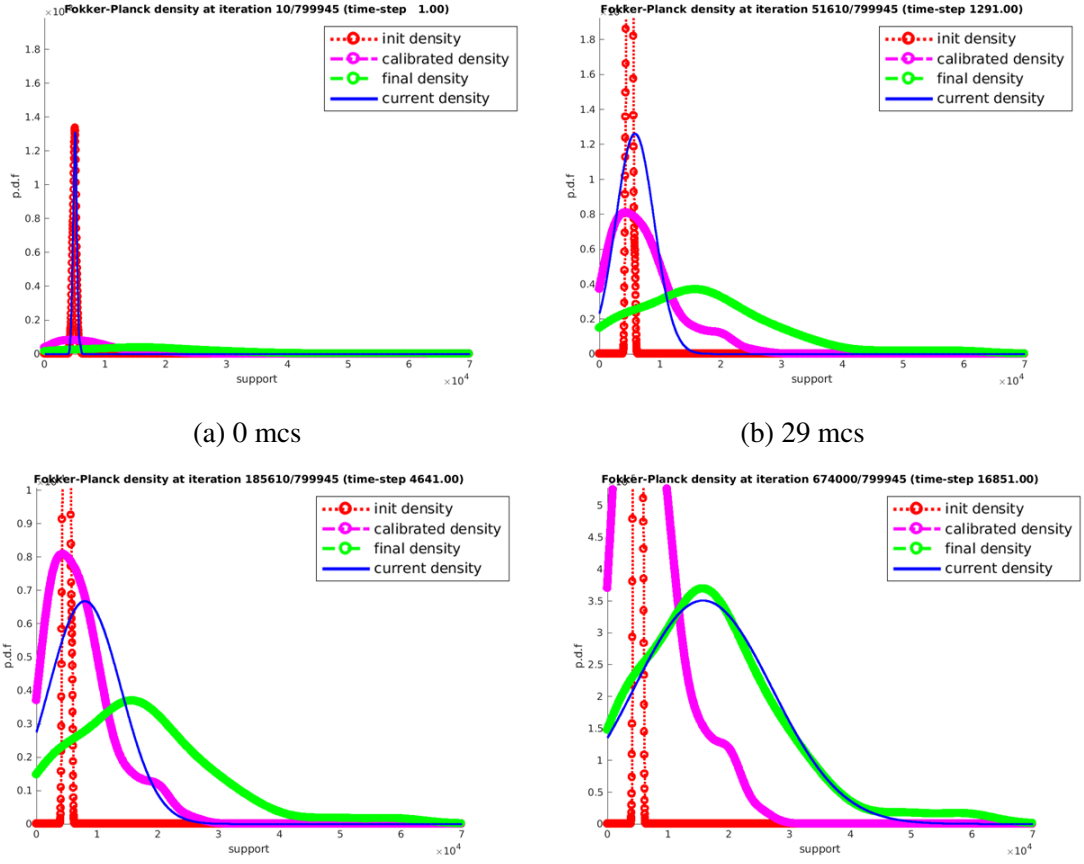
(a) 0 mcs (b) 29 mcs

(c) 34 mcs (d) 37 mcs

Figure 2.1: Microstructural evolution of grain growth in kMC simulation.

2.4.2 Phase field simulation: Spinodal decomposition

In this example, the microstructural evolution using PF simulation of Fe-Cr on MOOSE framework [46] is used to demonstrate the approach. In this example, the QoI is the chord-



(a) 0 mcs

(b) 29 mcs

(c) 34 mcs

(d) 37 mcs

Figure 2.2: Evolution of grain area as a function of time shows a good model's performance after calibration. (Readers are referred to online version to visualize different colors. Red curve denotes the initial PDF, magenta curve denotes the training PDF, green curve denotes the testing PDF, and the blue curve denotes the evolving Fokker-Planck PDF after calibration at different time steps.)

length distribution, which is another statistical microstructural descriptor. The training PDF and initial PDF are constructed using kernel density estimation method with the normal kernel distribution. The selected bandwidth is optimal for the normal kernel density [45]. The initial PDF is regularized using the Tikhinov regularization as described in Section 2.3.3 to reduce the probability of divergent Fokker-Planck solution.

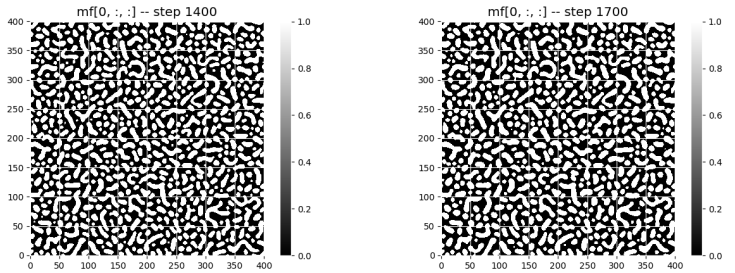
Figure 2.3 shows the microstructural evolution of Fe-Cr spinodal decomposition simulations on 25 nm $\times$ 25 nm at 500°C over a period of 7 days (604800s) in physical time. The system is modeled using Cahn-Hilliard equation with no external energy sources. The initial concentration of Cr is randomized on the interval [44.774%, 48.774%] with the expectation of 46.774%. The coarsening effect is observed, and the clusters slowly expand as the simulation advances.

The coefficients of the Fokker-Planck equation are calibrated using batch-parallel Bayesian optimization. Here, the drift and diffusion coefficients are parameterized, and the batch-parallel Bayesian optimization is applied to minimize the Kullbeck-Leibler divergence between the training PDF and the simulated Fokker-Planck PDF, as $\mathrm{KL}(p_{\text{calibrated}}||p_{\text{predicted}})$, where the $p_{\text{predicted}}$ is obtained from solving the forward Fokker-Planck equation with certain coefficients.

Figure 2.4 presents the evolution of calibrated Fokker-Planck equation to capture the evolution of QoI. Figure 2.3d shows the comparison between the PDF obtained by calibrated and trained Fokker-Planck equation and the testing PDF from the ICME model, which is the PF simulation in this case.

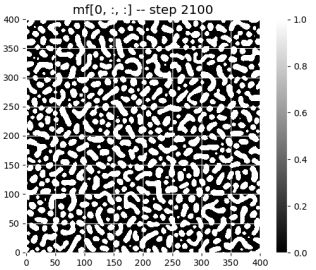
2.4.3 Molecular dynamics simulation: Equilibrium liquid Argon

In this example, MD simulation of liquid Argon at 85K is performed using LAMMPS framework [47] to assess the QoIs, which is the total mean square displacement and enthalpy of the simulation system. The system consists of 4000 atoms, where the interatomic potential is described by Lennard-Jones model with uncertain well-depth ε and

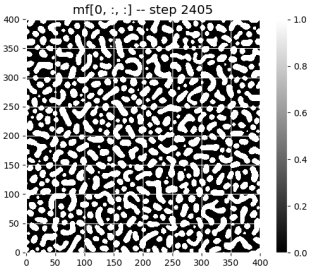


(a) 1400 steps

(b) 1700 steps



(c) 2100 steps

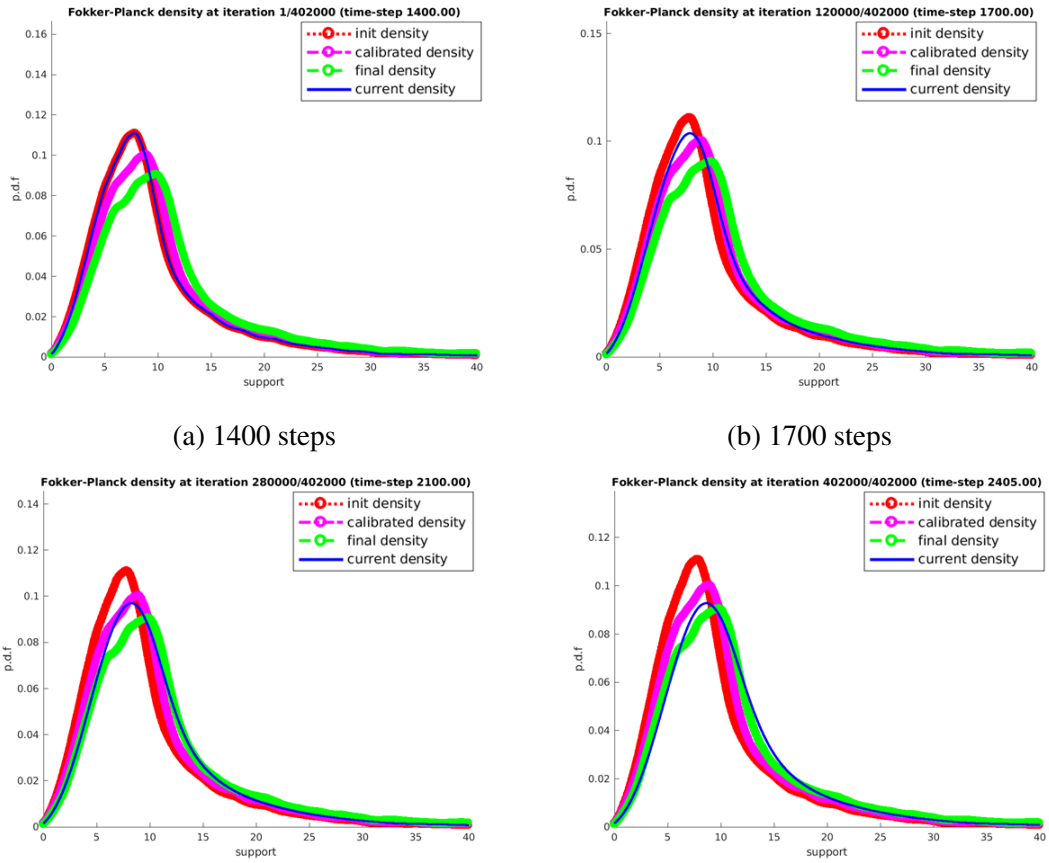


(d) 2405 steps

Figure 2.3: Microstructural evolution in PF spinodal decomposition simulation. There are two phases in this system: the Fe-rich and Cr-rich phases.

well-location σ. Different ε and σ for Argon have been used in the literature, for example, McGaughey et al. [48], Borgelt et al [49], Dawid et al. [50], Laasonen et al. [51], Reith et al. [52], Griebel et al. [53]. The uncertain ε and σ here is modeled with truncated normal distribution. The mean, the variance, the support lowerbound, and the support upperbound are (0.2383,0.0667,0.2376,0.2390) for ε, (3.4000,0.6670,3.3000,3.5000) for σ, respectively. The microcanonical ensemble (NVE) is used, where a Langevin thermostat is also used to coupled with the system. The Langevin thermostat, which has a random noise generator [54], can be thought of a source for aleatory uncertainty. The QoIs are analyzed using log files of the simulation, where the sampling time is 50 fs, the time step is 1 fs, and the total simulation time is 20 ps.

The uncertainty of the QoIs can be considered as a forward UQ problem, where the uncertainty is captured using PDF. Monte-Carlo sampling is used to assess the *a posteriori* distribution of the QoIs. The training PDF and initial PDF are reconstructed using kernel



(a) 1400 steps

(b) 1700 steps

(c) 2100 steps

(d) 2405 steps

Figure 2.4: Evolution of chord-length distribution shows a good agreement between the testing PDF from ICME and the PDF from the trained ROM. (Readers are referred to online version to visualize different colors. Red curve denotes the initial PDF, magenta curve denotes the training PDF, green curve denotes the testing PDF, and the blue curve denotes the evolving PDF after calibration at different time steps.)

density estimation method with the normal kernel distribution. The selected bandwidth is optimal for the normal kernel density [45]. The initial PDF for total mean-square displacement is regularized using the Tikhinov regularization as described in Section 2.3.3 to reduce the probability of divergent Fokker-Planck solution.

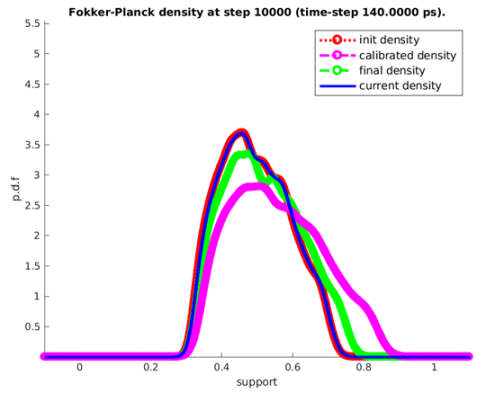
Figure 2.5 and Figure 2.6 show the evolution of the QoIs' PDFs in different snapshots at 140, 160, 180, and 200 ps. We note that in Figure 2.6, all the PDFs are fitted to a normal distribution. The Fokker-Planck coefficients for the total mean-square displacement are trained by minimizing the Kullbeck-Leibler divergence, whereas the coefficients for enthalpy are trained using Theorem 2 and Theorem 3. The comparison between the testing PDF and evolved Fokker-Planck PDF shows a fairly good agreement after the Fokker-Planck coefficients are calibrated.

2.5 Discussion

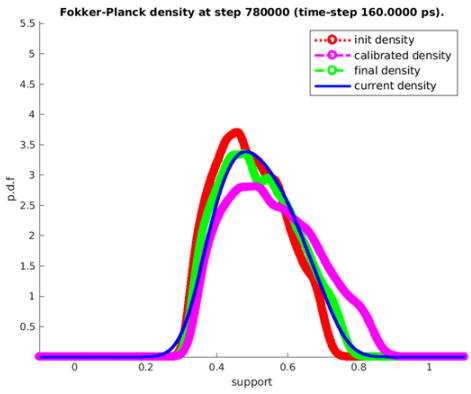
In this chapter, we present a time-upscaling stochastic ROM that leverage time-scale issue of ICME models, by approximating the evolution of QoIs using calibrated Fokker-Planck equation. ICME models are often limited by a particular time-scale since they are technically integrators, and parallelizing by spatial decomposition are easier compared to parallel-in-time.

The proposed method is demonstrated by both drift-dominated (as in the cases of MD and kMC) and diffusion-dominated (as in the case of PF) examples. It is shown that if the Fokker-Planck equation coefficients are appropriately parameterized, and the calibration for estimating the coefficients is performed correctly, the proposed method has a predictive capacity to estimate the QoI distribution in long run, without using the heavy computational ICME models.

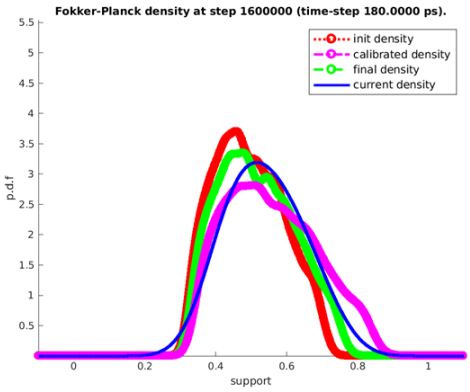
Building ROM for many QoIs, and coupling QoIs require a high-dimensional Fokker-Planck equation, which will be a topic of future study. We note that the QoIs can also be used to reconstruct the microstructure obtained by ICME models. For example, statistical



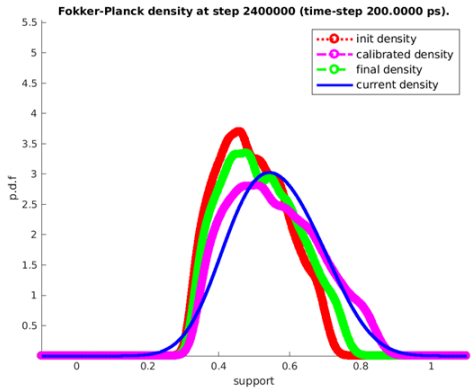
(a) 140 ps



(b) 160 ps

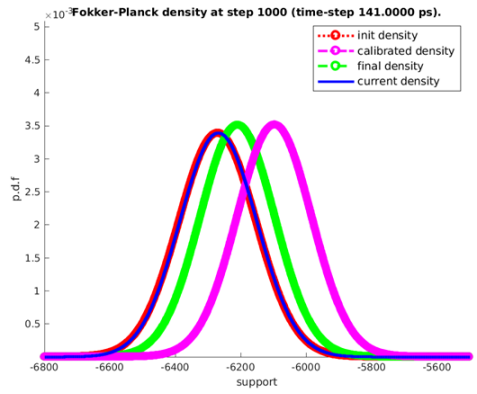


(c) 180 ps

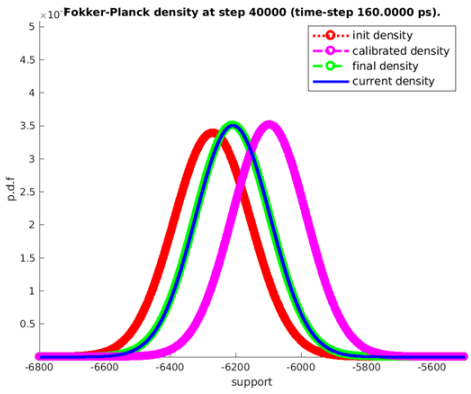


(d) 200 ps

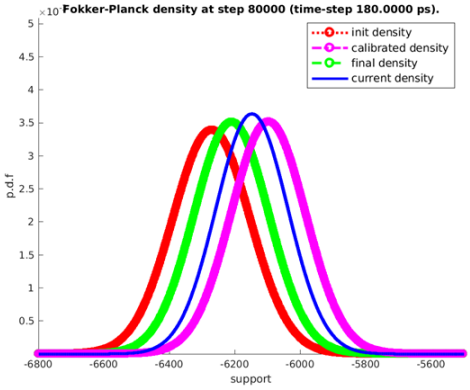
Figure 2.5: Evolution of total mean-square displacement in MD simulation. (Readers are referred to online version to visualize different colors. Red curve denotes the initial PDF, green curve denotes the training PDF, magenta curve denotes the testing PDF, and the blue curve denotes the evolving Fokker-Planck PDF after calibration at different time steps.)



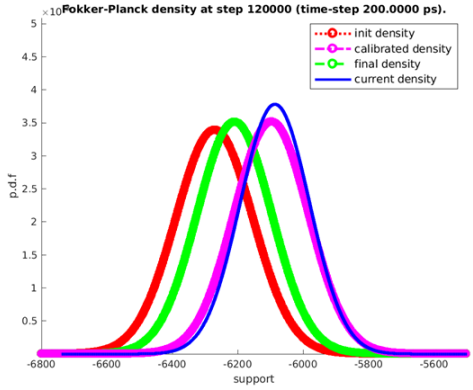
(a) 140 ps



(b) 160 ps



(c) 180 ps



(d) 200 ps

Figure 2.6: Evolution of enthalpy in MD simulation. (Readers are referred to online version to visualize different colors. Red curve denotes the initial PDF, green curve denotes the training PDF, magenta curve denotes the testing PDF, and the blue curve denotes the evolving Fokker-Planck PDF after calibration at different time steps.)

and deterministic microstructural descriptors are discussed in Torquato et al. [55] and Groeber et al. [3] [56], Chen et al. [2] [57]. It is noted that for computational materials science, the number of the statistical descriptors tend to dominate that of deterministic, due to the random nature of materials. The proposed framework in this chapter aims to evolve any of these statisticatical microstructural descriptors, as QoIs, for a longer time-scale that is not achievable by ICME models.

Propagating PDF along with time dimension is indeed more approachable, compared to the methods that propagate the moments and reconstruct the PDF based on the moments. For example, John et al. [58] reviews the literature for a number of methods and explains the ill-posed inverse problem. The Kramers-Moyal expansion allows one to propagate higher-order moments in theory, but in practice, estimations of the Kramers-Moyal coefficients are also more approachable using optimization methods. Furthermore, higher-order derivative terms make it harder to numerically solve for the forward Kramers-Moyal expansion. The proposed method is extensible in two directions. The former extension includes more variables in a high-dimensionality Fokker-Planck equation, whereas the later extension would capture the QoIs evolution with smaller approximation errors by including high-order terms for the same QoI. A caveat of the later extension is that with higher-order derivatives, it is likely that the numerical solution of the forward Kramers-Moyal expansion would diverge at some points, and some numerical treatments are required in order to obtain a convergent solution. One of those is the Tikhonov regularization method. In the other hand, one can also apply more complicated integration methods to stabilize and accelerate the integration in Kramers-Moyal expansion.

2.6 Summary

In this chapter, we propose a time-upscaling stochastic ROM to leverage the time-scale issue of ICME models. The ROM coefficients are trained either analytically or numerically, so that the evolution of QoI can be accurately captured using ROM. The main idea

of the chapter is to consider the QoI as a random variable, and propagate it using some stochastic processes. In this case, the Ornstein-Uhlenbeck stochastic process is modeled using 1D generalized Langevin equation. Under the formulation of Stratonovich calculus, the stochastic variable can be modeled using Fokker-Planck equation.

Three ICME examples are used to demonstrate the time-upscaling stochastic ROM framework, including kMC, PF, and MD, where the statistical microstructural descriptors are QoIs. The results show an excellent agreement between the trained ROM and the ICME prediction, as the proof-of-concept for the accurate prediction capacity of the ROM, assuming that it is parameterized and calibrated correctly.

CHAPTER 3

UNCERTAINTY QUANTIFICATION AND SENSITIVITY ANALYSIS OF PROCESS-STRUCTURE RELATIONSHIP IN AL-CU SOLIDIFICATION DENDRITE GROWTH

Phase field method (PFM) is a powerful simulation tool to predict the microstructural evolution as a function of thermodynamic and processing parameters. Different thermodynamic and processing parameters lead to different microstructure, and eventually to different materials properties. Therefore, it is important to explore the effects of these parameters upon the dendritic growth so that the process-structure relationship can be established. In PFM, the thermodynamic parameters can be obtained either through computational thermodynamics method, e.g. CALPHAD, or through ab-initio calculation. Both of these methods rely on some empirical assumptions, and thus result in uncertainty of the thermodynamic parameters. Furthermore, not every process can be perfectly controlled, thus the processing parameters are also uncertain. In this chapter, an uncertainty quantification (UQ) study of process-structure relationship is performed on the PFM to investigate the effects of the thermodynamic and processing parameters. To mitigate the curse-of-dimensionality and alleviate the computational burden, a sparse grid (SG) approach is adopted to interpolate four quantities of interests (QoIs), as a function of both thermodynamic and processing parameters. Image processing techniques are adopted to carefully examine the dendritic QoIs. The dendritic morphology is shown to vary significantly with respect to the interface mobility M_ϕ, which is a thermodynamic parameter, and the initial temperature T_0, which is a processing parameter. Qualitative and quantitative analyses on the QoIs are performed to provide computational insights on the dendritic growth.

3.1 Introduction

Phase field method (PFM) is a powerful simulation tool for studying the evolution of microstructure. The microstructure evolution, or more specifically the dendritic growth, can be modeled and captured as time evolves. From the computational materials science perspective, particularly for PFM, UQ plays a critical role in modeling materials structure at mesoscale. The reasons are three-fold. First, there exists spatially an inherent randomness in any representative volume element, which is the manifestation of the aleatory uncertainty at microscale level in materials science. Second, as PFM often relies on computational thermodynamics methods, such as CALPHAD [59, 60, 61, 62, 63], or *ab-initio* calculation for phase equilibria thermodynamic parameters [64, 65, 66, 67], there is another source of uncertainty involved during the integration process, which is usually referred to as epistemic uncertainty. For CALPHAD methods, the epistemic uncertainty originates from the inevitable experimental measurement and the curve fitting errors. For *ab-initio* calculations, the epistemic uncertainty is rooted at the approximation of potential energy landscape. Depending on the level of approximation, i.e. the rung of the so-called "Jacob's ladder" of approximation [68], the epistemic uncertainty can vary. Either way, the epistemic uncertainty is contained within these thermodynamic parameters, which are then used as inputs for the PFM. Third, any variation in the processing parameters, which are typically controllable, or thermodynamic parameters, which are uncontrollable, results in a different microstructure, and thus has a different property.

The microstructure is the key to investigate the process-structure-property relationship, because the microstructure links between processing parameters and the final materials properties. To study this chain relationship, PFM provides a computational tool that maps from the processes to the microstructure. However, the inputs of the PFM include not only processing parameters, but also the thermodynamic parameters. Thus, it is crucial to study the microstructural variation with respect to both the processing and thermodynamic

parameters.

There are some relevant work on the sensitivity analysis of phase field. Xing et al. [69] investigated the columnar dendritic growth competition with respect to the orientation in Al-4wt%Cu alloy. Takaki et al. [70] proposed a coupled PF-lattice Boltzmann model to study the Al-4wt%Cu for dendritic growth with natural convection. Qi et al. [71] studied the effects of the natural convection and solid motion on dendrite growth of the Al-4wt%Cu alloy using PFM and computational fluid dynamics (CFD). Liu and Wang [72] proposed a framework called phase field and thermo lattice Boltzmann model (PF-TLBM) to investigate the effects of cooling rate and latent heat on the dendrite growth of Al-4wt%Cu alloy. Boukellal et al. [73] investigated the growth of Al-4wt%Cu dendrites, comparing experiments with PFM, and proposed a scaling laws for the QoIs of tip location, tip velocity, and tip radius of curvature. However, the combined effects of thermodynamic and processing parameters upon the dendritic morphology are not well studied. More importantly, the uncertainty of the process-structure relationship for dendritic growth has not been quantified in the above work.

In this chapter, we study the effects of processing and thermodynamic parameters on the dendritic area S_d, the dendritic perimeter P_d, the segregation of Cu at solid-liquid interface κ, and the length of primary arm of dendrite P_d for Al-4wt%Cu binary alloy. An image processing pipeline is utilized and automated to investigate the effects of processing parameters, and quantify the variations of the four aforementioned QoIs. The SG method is used to interpolate these QoIs in high-dimensional input space and to leverage the computational burden for performing a large number of PFM. The dendritic morphology is presented on the node of SG, clearly demonstrating the parameters on the dendritic growth. The dendritic morphology is shown to vary significantly with respect to the interface mobility M_ϕ, which is a thermodynamic parameter, and the initial temperature T_0, which is a processing parameter. PCE framework is employed to quantify the QoIs, where the processing parameters are assumed to be deterministic and controllable, whereas the thermodynamic

parameters are assumed to be random. The UQ study provides computational insights on the dendritic growth, cementing the bridge between process-structure relationship for Al-4wt%Cu binary alloys.

In this chapter, Section 3.2 summarizes the formulation of SG for high-dimensional interpolation and integration method, and briefly discusses the formulation of PCE framework in a UQ problem. Section 3.3 describes the details about the PFM to study the dendritic evolution. Section 3.4 describes the automatic post-processing pipeline, in which four physical QoIs are extracted and studied as a function of thermodynamic and processing parameters. Section 3.5 analyzes the UQ results for two separate cases. In the first cases (Section 3.5.1), both the thermodynamic and processing parameters are treated as deterministic, and the QoIs are then represented as a high-dimensional interpolation quantities. In the second case (Section 3.5.2), the thermodynamic parameters are treated as random inputs, whereas the processing parameters are treated as deterministic parameters. Section 3.6 discusses the computational insights from the study, and Section 3.7 concludes the chapter.

3.2 Uncertainty quantification methodologies

3.2.1 SG high-dimensional interpolation

To mitigate the curse of dimensionality, the SG technique is employed to interpolate the QoIs on high-dimensional domain with d dimensionality. We follow the formulation of Barthelmann et al. [74] and Novak and Ritter [75] for the SG interpolation, in which global Lagrange polynomials are used as basis functions. For the SG method to be accurate, the interpolating function is typically assumed to be smooth. Accuracy of SG method has been studied extensively by Bungartz et al. [76] and Nobile et al [77].

Let f be the function (QoI) to be interpolate, for univariate interpolation,

$$\mathcal{U}^i(f) = \sum_{j=1}^{m_i} f(x_j^i) \cdot a_j^i, \tag{3.1}$$

where $i \in \mathbb{N}$, $a_j^i \in C([-1,1])$, where a_j^i is the Lagrange polynomial of degree $(m_i - 1)$, and $x_j^i \in [-1,1]$. For $d > 1$ multivariate pointwise interpolation, the full tensor product formulas is defined as

$$\mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}(f) = \sum_{j_1=1}^{m_{i_1}} \cdots \sum_{j_d=1}^{m_{i_d}} f(x_{j_1}^{i_1}, \ldots, x_{j_d}^{i_d}) \cdot (a_{j_1}^{i_1} \otimes \cdots \otimes a_{j_d}^{i_d}). \tag{3.2}$$

The Smolyak formula is then used to construct a linear operator $\mathcal{A}(q,d)$ as the linear combinations of product formulas, such that the products with a relatively small number of knots are used, and the linear combination is chosen in such a way that the interpolation property for univariate cases $d = 1$ is preserved for multivariate cases $d > 1$.

For $i = 0, \mathcal{U}^0 = 0$ and define

$$\Delta_i = \mathcal{U}^i - \mathcal{U}^{i-1} \tag{3.3}$$

Let d be the dimensionality of the problem, $|\boldsymbol{i}| = i_1 + \cdots + i_d$ for $\boldsymbol{i} \in \mathbb{N}^d$, then Smolyak's algorithm can be described as

$$\mathcal{A}(q,d) = \sum_{|\boldsymbol{i}| \leq q} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_d}\right), \tag{3.4}$$

for integers $q \geq d$. Shown by Wasilkowski and Woźniakowski [78], the Smolyak's algorithm can also be written as

$$\mathcal{A}(q,d) = \sum_{q-d+1 \leq |\boldsymbol{i}| \leq q} (-1)^{q-|\boldsymbol{i}|} \cdot \binom{d-1}{q-|\boldsymbol{i}|} \cdot \mathcal{U}^{i_1} \otimes \cdots \otimes \mathcal{U}^{i_d}. \tag{3.5}$$

To compute the linear operators $\mathcal{A}(q,d)$, the functional evaluations are performed at the SG, which is defined by

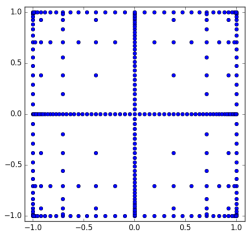
$$\mathcal{H}(q,d) = \bigcup_{q-d+1 \leq |\boldsymbol{i}| \leq q} \left(\mathcal{X}^{i_1} \times \cdots \times \mathcal{X}^{i_d}\right), \tag{3.6}$$

where $\mathcal{X}^i = \{x_1^i, \ldots, x_{m_i}^i\} \subset [-1,1]$ is the sets of nodes used by the linear operator $\mathcal{U}^i$.

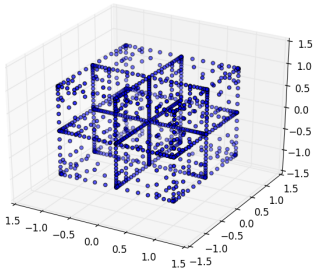
Root and extrema of Chebyshev polynomials are commonly used during the SG interpolation process. In this example, Chebyshev extremas are utilized according to the Clenshaw-Curtis rule [79] for the SG interpolation process as

$$x_j^i = -\cos\frac{\pi(j-1)}{m_i - 1}, j = 1, \ldots, m_i, \tag{3.7}$$

where $x_1^i = 0$ if $m_i = 1$, and $m_i = 2^{i-1} + 1$ for $i > 1$. This choice leads to a nested sets of point, $\mathcal{X}_i \subset \mathcal{X}_{i+1}$, and consequently, $\mathcal{H}(q,d) \subset \mathcal{H}(q+1,d)$, i.e. the lower-level grids are subsets of the higher-level grids. Figure 3.1a and Figure 3.1b show two examples of 2D and 3D SG, respectively, where the Clenshaw-Curtis nodes are used to construct the SG.



(a) $q = 6$, Clenshaw-Curtis nodes; $d = 2$; number of nodes per main dimension: 65; total number of nodes: 321



(b) $q = 6$, Clenshaw-Curtis nodes; $d = 3$; number of nodes per main dimension: 321; total number of nodes: 1073;

Figure 3.1: Example of SG in 2D (a) and 3D (b) for high dimensional interpolation.

3.2.2 Stochastic collocation for random input parameters

Polynomial chaos expansion (PCE) method is one of the most widely used UQ methods to quantify uncertainty in a system with random input parameters. In particular, generalized

polynomial chaos (gPC) expansion [80, 81] is an advanced and efficient tool, compared to other sampling methods. We adopt the notation from Xiu and Karniadakis [80] to briefly introduce the formulation of gPC. The gPC method seeks to represent the QoI, denoted as $u(\theta)$, as a general second-order process with finite variance, according to Askey scheme [82] of the generalized hypergeometric orthogonal polynomials, as

$$\begin{aligned} u(\theta) &= a_0 I_0 \\ &+ \sum_{i_1=1}^{\infty} c_{i_1} I_1(\zeta_{i_1}(\theta)) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty} c_{i_1 i_2} I_2(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta)) \\ &+ \sum_{i_1=1}^{\infty}\sum_{i_2=1}^{\infty}\sum_{i_3=1}^{\infty} c_{i_1 i_2 i_3} I_3(\zeta_{i_1}(\theta), \zeta_{i_2}(\theta), \zeta_{i_3}(\theta)) \\ &+ \cdots, \end{aligned} \tag{3.8}$$

where $I_n(\zeta_{i_1}, \cdots, \zeta_{i_n})$ denotes the Wiener-Askey polynomial chaos of order n in terms of the random vector $\boldsymbol{\zeta} = (\zeta_{i_1}, \zeta_{i_2}, \ldots, \zeta_{i_n})$. Without loss of generality, Equation 3.8 can be rewritten as

$$u(\theta) = \sum_{j=0}^{\infty} \hat{u}_j \boldsymbol{\Phi}_j(\boldsymbol{\zeta}), \tag{3.9}$$

where there is a one-to-one correspondence between the function $I_n(\zeta_{i_1}, \cdots, \zeta_{i_n})$ and $\boldsymbol{\Phi}_j(\boldsymbol{\zeta})$.

In practice, the number of terms in Equation 3.9 are truncated after a finite term P, where $P + 1 = \frac{(p+d)!}{p!d!}$, where p is the order of PCE, and d is the dimensionality of the problem, resulting in an approximation for finite PCE,

$$u(\theta) \approx \sum_{j=0}^{P} \hat{u}_j \boldsymbol{\Phi}_j(\boldsymbol{\zeta}), \tag{3.10}$$

The PCE coefficients $\hat{u}_j$ is calculated by the Galerkin projection of the Equation 3.10 onto the orthogonal polynomial basis $\{\boldsymbol{\Phi}_j\}$, as

$$\hat{u}_j = \frac{\langle u \boldsymbol{\Phi}_j \rangle}{\langle \boldsymbol{\Phi}_j^2 \rangle}. \tag{3.11}$$

The PCE coefficients can be obtained in multiple ways, by evaluating the high-dimensional integrals in the numerators and denominators of Equation 3.11. For high-dimensional problem, the computational cost to evaluate the integrals becomes prohibitive. Stochastic collocation method mitigates the curse-of-dimensionality on the evaluation of integrals through SG [83, 84, 85, 86, 77].

3.3 Phase field simulation

Multi-phase multi-component field is a generic formulation for phase transition of alloys. In this work, multi-phase field method described in [87] is adopted. The essential component of PFM is a free energy functional that describes the kinetics of phase transition. The free energy functional

$$F = \int_{\Omega} (f^{\text{GB}} + f^{\text{CH}}) dV \tag{3.12}$$

is defined with an interfacial free energy density f^{GB} and a chemical free energy density f^{CH} in a domain Ω.

A continuous variable named phased field ϕ indicates the fraction of solid phase in the simulation domain during the solidification process, and the fraction of liquid phase is $\phi_l = 1 - \phi$. The interfacial free energy density is defined as

$$f^{\text{GB}} = \frac{4\sigma^*(\boldsymbol{n})}{\eta} \left\{ |\nabla\phi|^2 + \frac{\pi^2}{\eta^2}\phi(1-\phi) \right\}, \tag{3.13}$$

where $\sigma^*(\boldsymbol{n})$ is the anisotropic interfacial energy stiffness, η is the interfacial width, $\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}$ is the local normal direction of the interface. The anisotropic interfacial energy stiffness is defined as

$$\sigma^* = \sigma + \frac{\partial^2\sigma}{\partial\theta^2} = \sigma_0^*[1 - 3\varepsilon^* + 4\varepsilon^*(n_x^4 + n_y^4)], \tag{3.14}$$

where σ is the interfacial energy, $\theta = \text{atan}\left(\frac{n_y}{n_x}\right)$ indicates the orientation, σ_0^* is the prefactor

of interfacial energy stiffness, and ε^* is the anisotropy strength of interfacial energy stiffness, which models the difference between the primary and secondary growth directions of dendrites.

The chemical free energy is the combination of bulk free energies of individual phases as

$$f^{\text{CH}} = h(\phi) f_s(C_s) + h(1-\phi) f_l C_l + \mu[C - (\phi_s C_s + \phi_l C_l)], \tag{3.15}$$

where C_s and C_l are the compositions of solutions as weight percent (wt%) in solid and liquid phases respectively, which is the amount of solute dissolved in a specific amount of solution. C is the overall composition of a solution in the simulation domain. $f_s(C_s)$ and $f_l(C_l)$ are the chemical bulk free energy densities of solid and liquid phases respectively. μ is the generalized chemical potential of solute introduced as a Lagrange multiplier to conserve the solute mass balance $C = \phi_s C_s + \phi_l C_l$. The weight function

$$h(\phi) = \frac{1}{4}\left[(2\phi - 1)\sqrt{\phi(1-\phi)} + \frac{1}{2}\text{asin}(2\phi - 1)\right] \tag{3.16}$$

provides the coefficients associated with solid and liquid bulk energies.

The evolution of the phase field is described by

$$\dot{\phi} = M_\phi \left\{ \sigma^*(\boldsymbol{n}) \left[\nabla^2 \phi + \frac{\pi^2}{\eta^2}(\phi - \frac{1}{2}) \right] + \frac{\pi}{\eta}\sqrt{\phi(1-\phi)}\Delta G \right\}, \tag{3.17}$$

where M_ϕ is the coefficient of interface mobility, and the driving force is given by

$$\Delta G = \Delta S(T_m - T - m_l C_l), \tag{3.18}$$

where $\Delta S = -1 \cdot 10^6 J \cdot K^{-1}$ is the entropy difference between the solid and liquid phase, T_m is the melting temperature of a pure substance, T is the temperature field, and m_l is slope of liquidus. For simplification, the interface mobility is assumed to be constant in this work.

The evolution of composition variable is modeled by

$$\dot{C} = \nabla \cdot [D_l(1-\phi)\nabla C_l] + \nabla \cdot \boldsymbol{j}_{\text{at}}, \tag{3.19}$$

where $k = \frac{C_s}{C_l}$ is the local partition coefficient and D_l is the diffusion coefficient of liquid. Furthermore, $\boldsymbol{j}_{\text{at}}$ is the anti-trapping current and defined as

$$\boldsymbol{j}_{\text{at}} = \frac{\eta}{\pi}\sqrt{\phi(1-\phi)}(C_l - C_s)\dot{\phi}\frac{\nabla\phi}{|\nabla\phi|} \tag{3.20}$$

which is to eliminate the unphysical solute trapping during the interface diffusion process by removing the anomalous chemical potential jump [88, 89] so that simulations can be done more efficiently with the simulated interface width larger than the physical one. Equation 3.17 and Equation 3.19 are the main equations to solve during PFM.

The open-source PFM toolkit OpenPhase [90] is used to simulate the two dimensional (2D) dendritic growth of binary alloy Al-4wt%Cu. Table 1 shows the physical properties of Al-4wt%Cu alloy. In all simulation runs, the grid spacing is $\Delta x = 1\mu$m, the time step is $\Delta t = 2 \cdot 10^{-5}$s, and the simulation period is 0.12 s. The length and width of the simulated domain are $L_x = 300\mu$m and $L_y = 300\mu$m in x- ($< 1, 0, 0 >$) and y-directions ($< 0, 1, 0 >$), respectively. The initial diameter of the seed is $D = 10\mu$m, and the width of interface is $\eta = 5\mu$m. The initial position of the seed is at the center of the simulation box. The initial concentration of Cu is $C_0 = 4$wt% for the liquid melt. For phase field ϕ and concentration C, zero Neumann conditions are set at all boundaries.

Table 3.1 lists the physical parameters of the Al-4wt%Cu alloy concerned in the study. The dendritic growth of Al-4wt%Cu is investigated through a parametric study of PFM, where the input parameters are varied, and the QoIs are investigated using an automatic post-processing pipeline. Table 3.2 describe the range of input parameters and their physical meaning in the PFM study.

Table 3.1: The physical properties of Al-4wt%Cu alloy.

Symbol	Physical meaning	Value	Unit
T_m	Melting point of pure Al	933.6	K
ρ	Density of liquid	2700	kg/m^3
m_l	Slope of liquidus	-2.6	K/wt%
k	Partition coefficient	0.14	
D_l	Diffusivity of liquid	$3.0 \cdot 10^{-9}$	m^2/s
σ_0^*	Prefactor of interfacial energy stiffness	0.24	J/m^2
ε^*	Interfacial energy stiffness anisotropy	0.35	
M_ϕ	Interface mobility	$4 \cdot 10^{-9}$	m^4/(J$\cdot$ s)

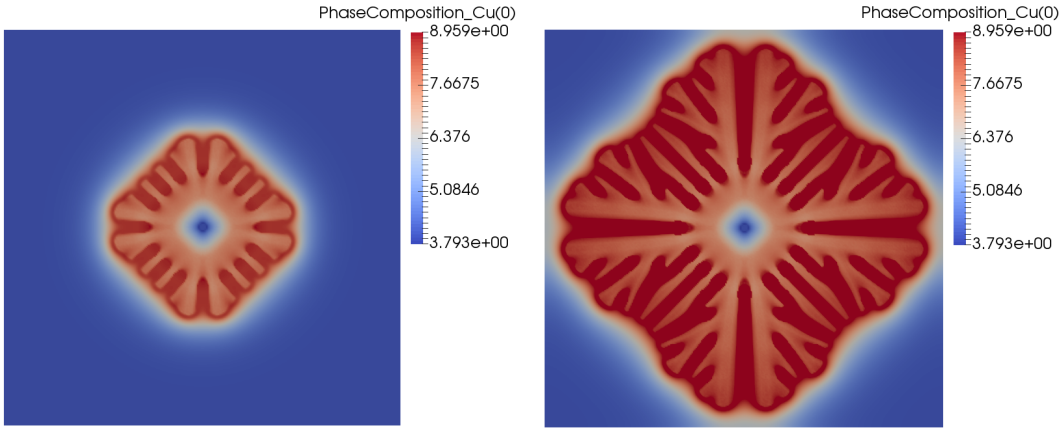
Table 3.2: Input parameters and its according range.

Symbol	Physical meaning	Lower bound	Upper bound	Unit
$\frac{\partial T}{\partial t}$	Cooling rate	-20	-10	K/s
T_0	Initial temperature	915	920	K
σ_0^*	Prefactor of interfacial energy stiffness	0.22	0.26	J/m^2
ε^*	Interface anisotropy	0.30	0.40	
M_ϕ	Interface mobility	$3 \cdot 10^{-9}$	$5 \cdot 10^{-9}$	m^4/(J$\cdot$ s)

3.4 Simulation procedure

Figure 3.2a and Figure 3.2b show an example of dendritic growth at different snapshot in the 2D PFM. The primary and secondary dendritic arms are observed in the PFM. The input parameters listed in Table 3.2 have direct impact on the geometry and shape of the dendritic growth. Thus, an automatic post-processing pipeline is devised to extract the QoIs that describe the dendrite.

A level-7 SG for 5-dimensional space is constructed based on Clenshaw-Curtis rule [79], where the nested Chebyshev nodes are chosen. Tasmanian package [91, 92, 93, 94] is used to construct and evaluate for high-dimensional interpolation. The construction of SG results in 19313 nodes, which corresponds to different set of inputs for the PFM. At each node, a PFM is performed, and the QoIs are collected once the simulation is finished. The process is automated using Bash/Shell and Python scripts. The mappings from the processing space to microstructure space and its descriptors (QoIs) are known to be smooth, as no singularity is expected for the PFM, and the dendrite is expected to evolve continuously as



(a) An example of dendritic growth at timestep 3000.

(b) An example of dendritic growth at timestep 6000.

Figure 3.2: Dendritic growth of binary alloy Al-4wt%Cu at different snapshots.

time advances.

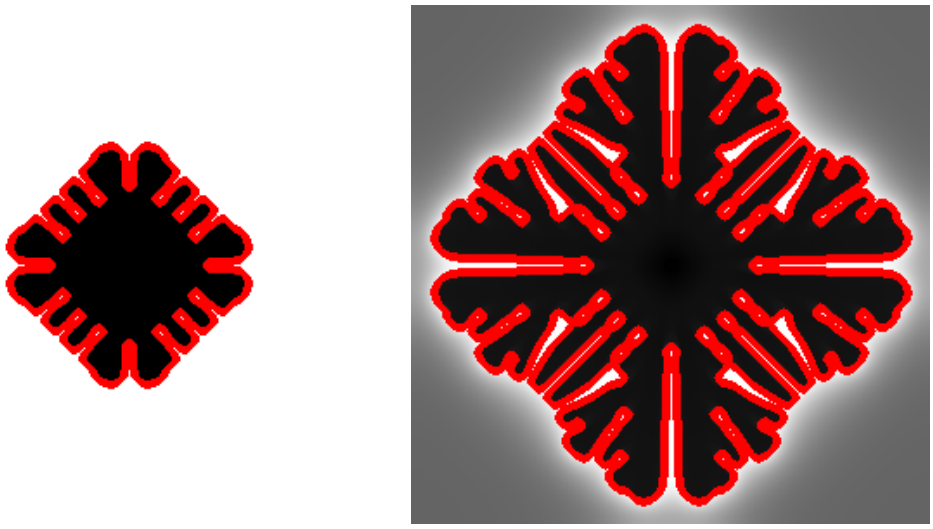
3.4.1 Quantities of interests

In this section, the QoIs in this UQ study are described in details. In this work, four QoIs are considered: (1) the perimeter of dendrite L_d, (2) the area of dendrite S_d, (3) the segregation of Cu at solid-liquid interface κ, and (4) the length of primary arm of dendrite P_d.

Dendritic perimeter

Figure 3.3a and Figure 3.3b present the contour of the dendrite in Al-4wt%Cu binary alloy at different snapshots. The contours are plotted as red wrapping around the solid dendrite.

The perimeter of the dendrite, denoted as L_d, is the first QoI. To compute the perimeter of the dendrite, the phase field composition is extracted after a number of timesteps. Readers are referred to the online manuscript for the color version of the figures. A threshold is imposed based on the phase field composition to convert the dendrite to gray scale image. Then, the contours are retrieved from the binary image using Suzuki algorithm [95]. The contour extraction is implemented based on the OpenCV open source code [96].



(a) Dendrite contour at timestep 3000. (b) Dendrite contour at timestep 6000.

Figure 3.3: Perimeter computation of the dendrite in Figure 3.2 via finding contour in image processing at different snapshots. Readers are referred to the online manuscript for color version. Red lines indicate the contours, whereas black region corresponds Cu-rich region, and white region corresponds to Al-rich region.

Dendritic area

The area of the dendrite, denoted as S_d, is the second QoI in this study. The computation of dendritic area is performed in a similar approach with the dendritic perimeter, in which a phase field contour is extracted based on the phase field composition function, as in Figure 3.3. After a non-self-intersecting contour of dendrite is extracted, the dendritic area is computed as the contour area through Green formula. The computation of dendritic area occurs after a fixed number of time steps, based on the frequency of phase field composition outputs of the PFM, and is implemented using the OpenCV package, similarly with the dendritic perimeter computation.

Cu segregation

The segregation of Cu at solid-liquid interface, denoted as κ, is the third QoI, and calculated as

$$\kappa = \frac{C_l^i}{C_s^i} \tag{3.21}$$

where C_l^i and C_s^i are the composition of liquid and solid phase at the interface, respectively. The computation of Cu segregation quantity occurs after a fixed number of timesteps, based on the phase field composition outputs of the PFM. It has been shown that high Cu segregation indicator κ promotes Al_2Cu θ intermetallic phase [97] on the grain boundary or inside the grain. Al_2Cu θ phase is associated with higher mechanical strength of the materials. Thus, κ serves as an implicit link between structure and property relationship.

Dendritic primary arm length

The dendritic primary arm length, denoted as P_d, is the fourth QoI. Based on the extracted spatial phase field composition of the PFM, the dendritic primary arm length is computed based on the location of the pixel in binary image, in which Cu-rich phase switches to Al-rich phase, along $< 1, 0, 0 >$ and $< 0, 1, 0 >$ crystallographic directions.

3.5 Numerical Results

In Section 3.5.1, the QoIs are explored as a function of input parameters, where both processing and thermodynamic parameters in Table 3.2 are treated as deterministic variables. The SG method is then employed to expand the QoIs as a high-dimensional interpolation of the input parameters.

In Section 3.5.2, we interpret the physical meaning of the PFM input parameters as follows. The processing parameters, T_0 and $\frac{\partial T}{\partial t}$, are treated as deterministic variables, whereas the thermodynamic parameters, σ_0^*, ε^*, and M_ϕ, are treated as random parameters, which are uniform distributed between the lower and upper bounds in Table 3.2. On one hand, in the context of mechanical engineering and manufacturing, the manufacturing conditions are often controlled very tight, and thus it is physically plausible to control and manipulate the manufacturing conditions, as needed. On the other hand, the thermodynamic parameters, which are often obtained through experiments or other ICME methods, such as CALPHAD [59, 60], *ab-initio* [68], the obtained parameters are not exact and prone to uncertainty in

other ICME tools.

3.5.1 SG: Interpolation with deterministic input parameters

Figure 3.4 and Figure 3.6 shows dendritic morphology variation on SG nodes, as a function of processing parameters and thermodynamic parameters, respectively. Low cooling rate and low initial temperature both appear to promote the growth of the dendrite secondary arms. The initial temperature seems to have a bigger impact on the secondary arm length than the cooling rate, as at $T_0 = 915K$, the secondary arms grow more densely, compared to those at $T_0 = 920K$. To be representative, the dendrite at each node is fixed at a nominal thermodynamic parameters. However, at each node, there is also another collection of dendritic morphology, where other thermodynamic parameters vary.

Figure 3.5 shows the contour map of four different QoIs as a function of processing parameters, where the dendritic morphology on SG is shown in Figure 3.4. The dendritic morphology changes dramatically with respect to the initial temperature. The low initial temperature promotes the dendritic growth in all aspects, particularly the dendrite secondary arm growth. The secondary arm counts, as well as the dendritic area and parameter are monotonic in the chosen bound of the initial temperature $[915, 920]K$. The cooling rate also has an effect on the dendritic growth. However, quantitatively, the dendritic morphology does not change significantly with respect to the cooling rate in the chosen bound [-20,-10] K/s, as shown in Figure 3.4. Qualitatively, higher cooling rate promotes dendritic growth, as manifested by the dendritic area and dendritic perimeter, as shown in Figure 3.5. However, the initial temperature appears to play a major role in promoting dendritic growth, as well as morphing the dendrite. Dendrites with more secondary arms have larger area and perimeter.

Cu segregation κ is a monotonic function of processing parameters, including cooling rate $\frac{\partial T}{\partial t}$ and the initial temperature T_0, in the range of study, as shown in Figure 3.5c. Particularly, the Cu segregation κ increases with respect to a faster cooling rate and lower

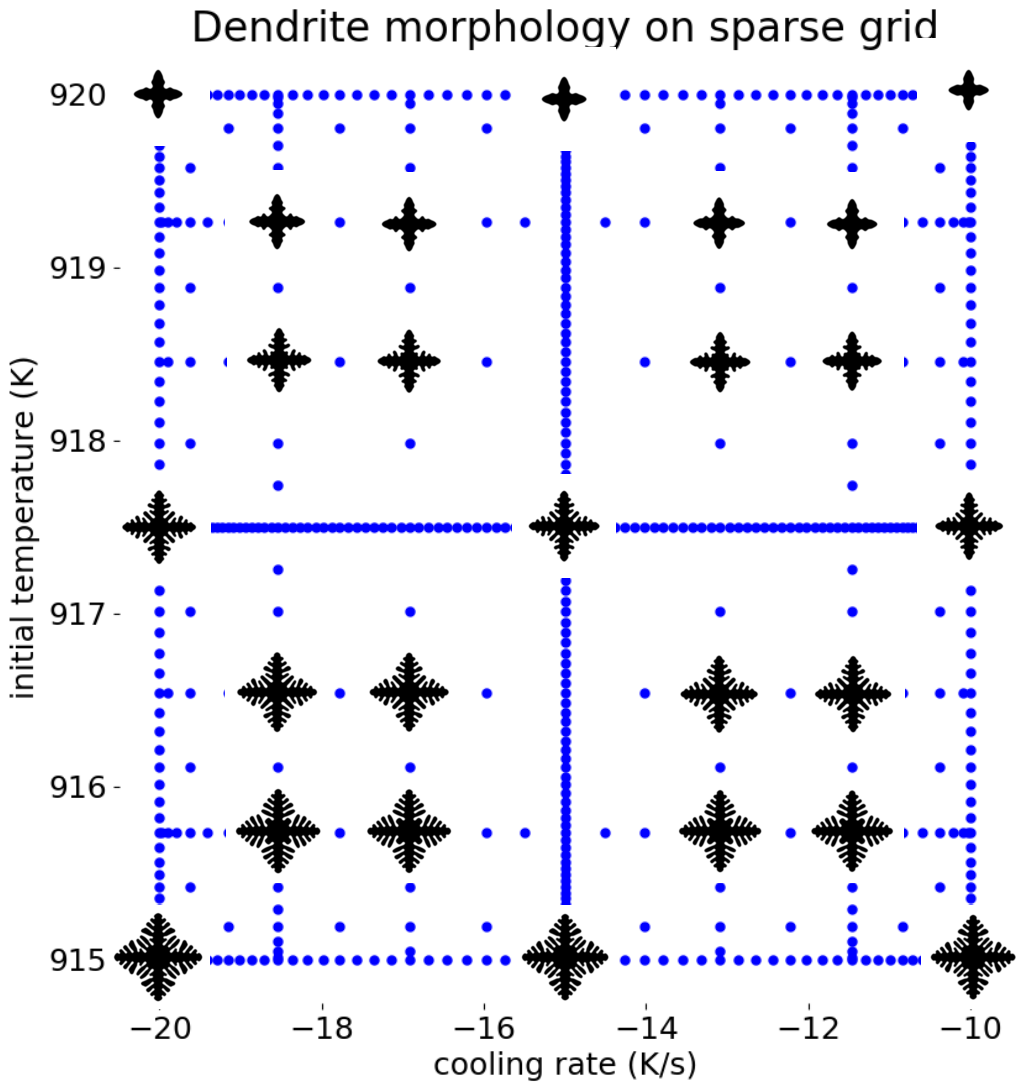
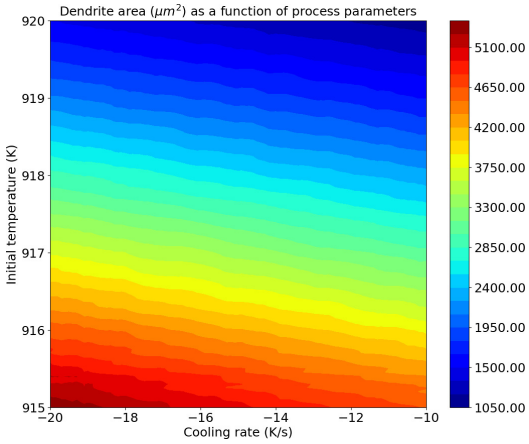
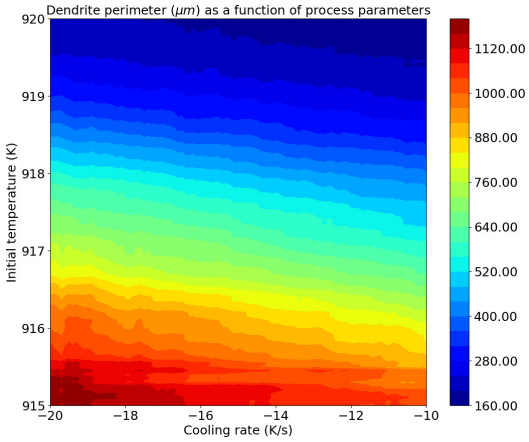


Figure 3.4: Representative dendritic morphology on SG as a function of process parameters, i.e. cooling rate and initial temperature. The thermodynamic parameters are fixed at $\varepsilon^* = 0.35, \sigma_0^* = 0.24, M_\phi = 4 \cdot 10^{-9}$, respectively.

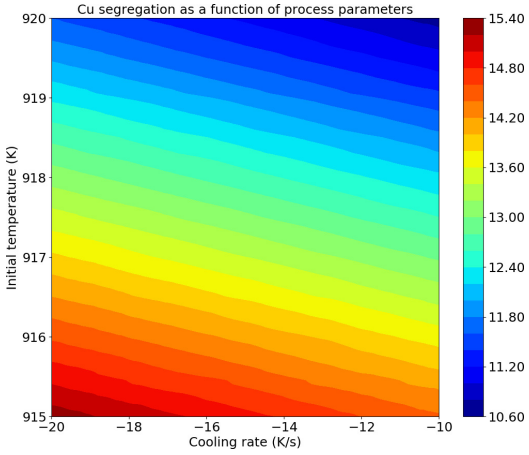
initial temperature. The initial temperature T_0 has a dominant effect on the Cu segregation κ for the range used in this chapter. The primary arm length P_d, as shown in Figure 3.5d is a highly nonlinear function of the processing parameters, but appears to be correlate with initial temperature as well. Lower initial temperature T_0 tends to weakly associate with higher primary dendritic arm length P_d.



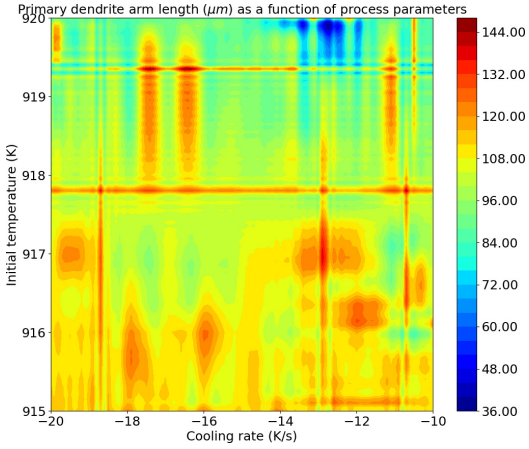
(a) dendritic area $(\mu m)^2$ as a function of processing parameters. Other thermodynamic parameters are fixed at $\varepsilon^* = 0.35, \sigma_0^* = 0.24, M_\phi = 4 \cdot 10^{-9}$, respectively.



(b) dendritic perimeter (μm) as a function of processing parameters. Other thermodynamic parameters are fixed at $\varepsilon^* = 0.35, \sigma_0^* = 0.24, M_\phi = 4 \cdot 10^{-9}$, respectively.



(c) Cu segregation as a function of processing parameters. Other thermodynamic parameters are fixed at $\varepsilon^* = 0.35, \sigma_0^* = 0.24, M_\phi = 4 \cdot 10^{-9}$, respectively.



(d) dendritic primary arm length (μm) as a function of processing parameters. Other thermodynamic parameters are fixed at $\varepsilon^* = 0.35, \sigma_0^* = 0.24, M_\phi = 4 \cdot 10^{-9}$, respectively.

Figure 3.5: Dendrite QoIs as a function of processing parameters.

Figure 3.6 shows the dendritic morphology variation with respect to the thermodynamic parameters, i.e. interface anisotropy ε^* and interface mobility M_ϕ. The dendritic morphology varies significantly with respect to the interface mobility M_ϕ. Higher interface mobility M_ϕ promotes the growth of dendrite secondary arms, and consequently, the dendritic area and the dendritic perimeter. It is noted that there is a small difference in Figure 3.6 and Figure 3.4 at the center of the dendrites. Particularly, the center of the dendrites in Figure 3.4 is more developed than the center of the dendrites in Figure 3.6. The center of the dendrites is appeared to impact the dendritic area and dendritic perimeter. If the center of the dendrites is well-developed, with substantial center secondary arm growth, then the dendritic perimeter and dendritic area increase. If the center of the dendrites is under-developed, then the dendritic area and dendritic perimeter decrease.

It is observed that the dendritic area S_d and the dendritic perimeter P_d significantly increase when the interface mobility M_ϕ increases. This is because a higher interface mobility results in a higher growth speed of dendrite. When the initial temperature is low (915K), the dendritic area and perimeter are positively correlated with the interfacial energy stiffness. When the initial temperature is high (920K), which means the undercooling is low, the change of interfacial energy stiffness has a trivial effect on the dendritic area and perimeter. The interface anisotropy ε^* does not change the dendritic area S_d and the dendritic perimeter P_d much. Quantitatively, the interface anisotropic ε^* parameter does not have a significant impact on the dendritic morphology. Qualitatively, as shown in Figure 3.7, the dendritic area and dendritic perimeter is a strong function of the interface mobility M_ϕ. A small increasing M_ϕ substantially promotes the dendritic growth.

As illustrated in Figure 3.7c, the Cu segregation κ generally increases as the interface mobility M_ϕ increases. The relationship between κ and M_ϕ is not strictly monotonic and noisy. However, comparing between Figure 3.7c and Figure 3.5c, the processing parameters have a stronger effect on the magnitude of Cu segregation than the thermodynamic parameters. The variation bound for κ in Figure 3.5c is [10.60,15,40], compared to

[12.74,13.28] in Figure 3.5c.

The dendritic primary arm length P_d is shown to be a nonlinear function of thermodynamic parameters, as in Figure 3.7d. However, the variation is fairly mild, as most of the dendrites achieve roughly the same tip location with different thermodynamic parameters. There is a weak positive correlation between the interface mobility M_ϕ and the dendritic primary arm length P_d.

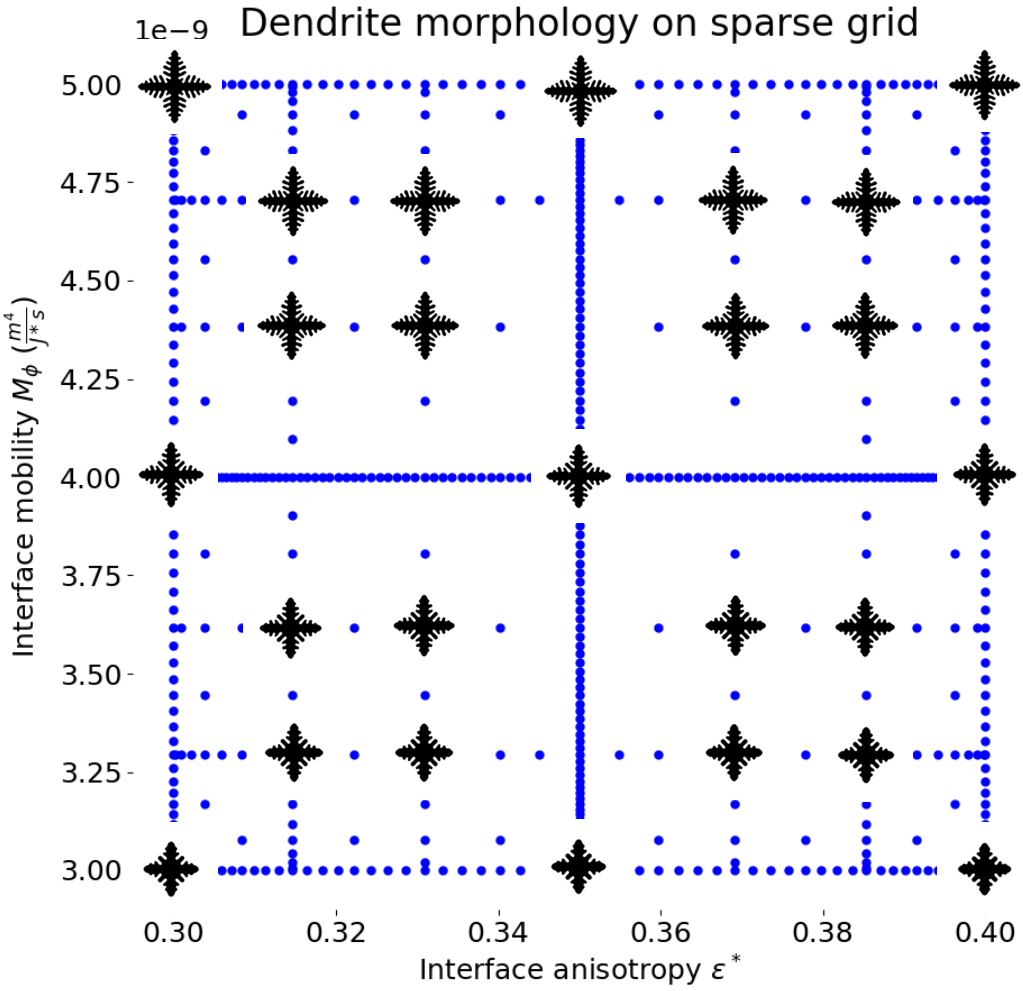
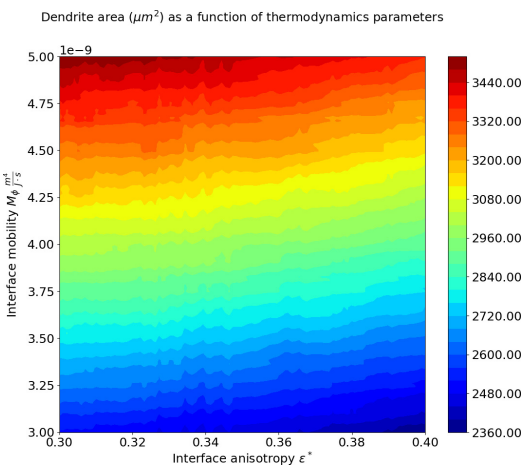
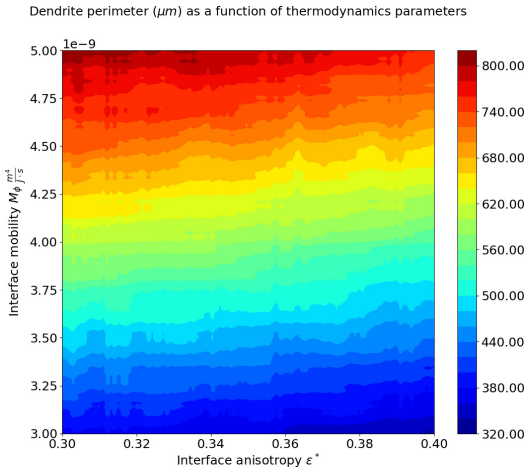


Figure 3.6: dendritic morphology at different parameters on SG, where other parameters are fixed at $\frac{\partial T}{\partial t} = -15K/s$, $T_0 = 917.5$, and $\sigma_0^* = 0.24$.

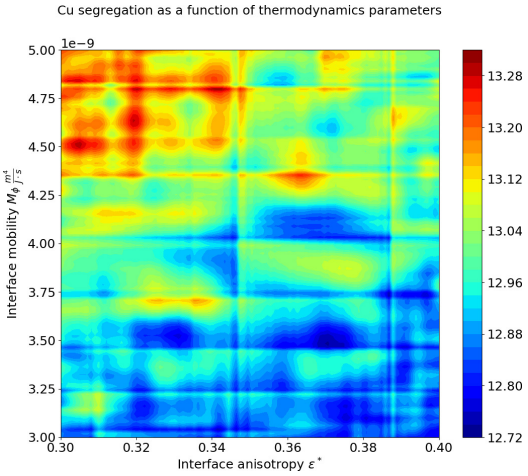
To further investigate the effect of all parameters, including processing and thermody-



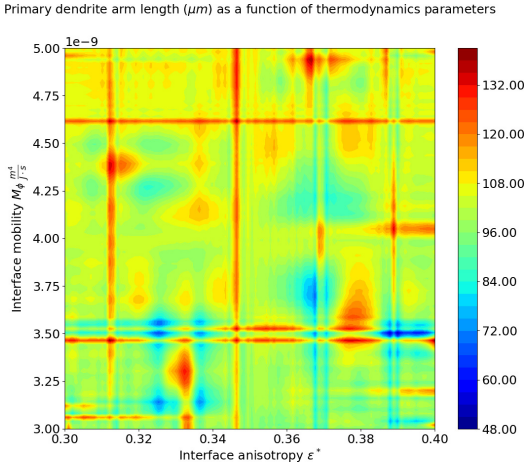
(a) dendritic area $(\mu m)^2$ as a function of thermodynamic parameters. Other thermodynamic parameters are fixed at $\frac{\partial T}{\partial t} = -15K/s$, $T_0 = 917.5$, and $\sigma_0^* = 0.24$.



(b) dendritic perimeter (μm) as a function of thermodynamic parameters. Other thermodynamic parameters are fixed at $\frac{\partial T}{\partial t} = -15K/s$, $T_0 = 917.5$, and $\sigma_0^* = 0.24$.



(c) Cu segregation as a function of thermodynamic parameters. Other thermodynamic parameters are fixed at $\frac{\partial T}{\partial t} = -15K/s$, $T_0 = 917.5$, and $\sigma_0^* = 0.24$.

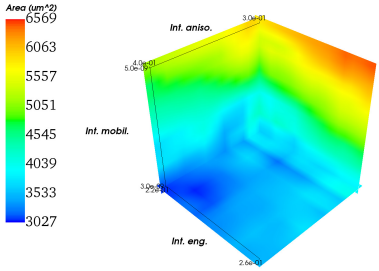


(d) dendritic primary arm length (μm) as a function of thermodynamic parameters. Other thermodynamic parameters are fixed at $\frac{\partial T}{\partial t} = -15K/s$, $T_0 = 917.5$, and $\sigma_0^* = 0.24$.

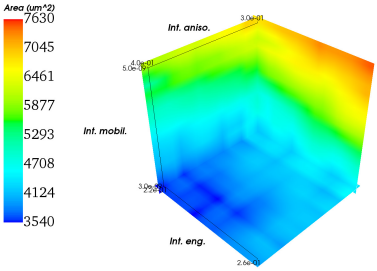
Figure 3.7: Dendrite QoIs as a function of thermodynamic parameters.

namic parameters on the dendrite, 3D contours of all QoIs, i.e. the dendritic area, dendritic perimeter, Cu segregation, and dendritic primary arm length, are plotted in Figure 3.8, Figure 3.9, Figure 3.10, and Figure 3.11, respectively. The processing parameters are further divided into subplot of each figure, where the cooling rate $\frac{\partial T}{\partial t}$ and the initial temperature T_0 are fixed at four corners of the SG, corresponding to the lower and upper bounds of each parameters. The QoIs are then plotted as a function of three thermodynamic properties, i.e. prefactor of interfacial energy stiffness σ_0^*, interface anisotropy ε^*, and interface mobility M_ϕ, using 3D contour plots. Readers are referred to the online version of color plots.

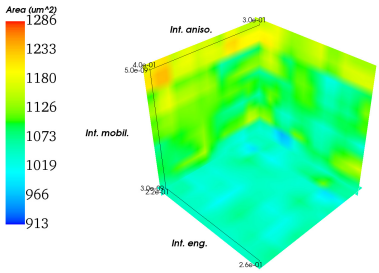
Figure 3.8 shows the dendritic area variation of processing and thermodynamic parameters. Initial temperature T_0 plays a major role in dendritic growth, as slightly lower temperature significantly accelerate dendritic area. An increase in interface mobility M_ϕ corresponds to an increase in dendritic area. Figure 3.9 shows the dendritic perimeter variations as a function of input parameters. The dendritic area and dendritic perimeter are highly correlated, as shown in Figure 3.8 and Figure 3.9. Figure 3.10 plots the Cu segregation κ, showing a mild variation with respect to thermodynamic parameters. Namely, lower interface anisotropy ε^* and higher interface mobility M_ϕ generally result in higher κ. However, as shown previously in Figure 3.5c, the Cu segregation κ is more sensitive to the initial temperature T_0, compared to other thermodynamic parameters. It has been shown that high κ promotes θ phase Al_2C on the grain boundary or inside the grain, consequently resulting in a higher mechanical strength of material. Figure 3.11 presents the dendritic primary arm length, showing a mild dependence on the initial temperature T_0 and the interface mobility M_ϕ, as in Figure 3.5d and Figure 3.7d, respectively.



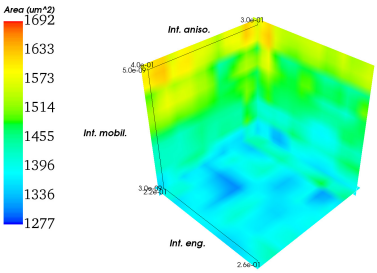
(a) dendritic area as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 915K, respectively.



(b) dendritic area as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 915K, respectively.

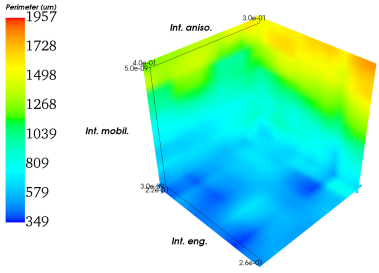


(c) dendritic area as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 920K, respectively.

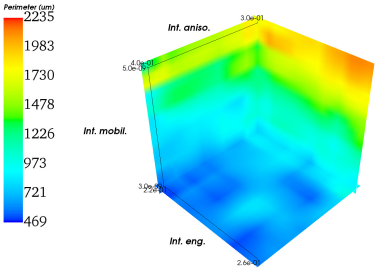


(d) dendritic area as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 920K, respectively.

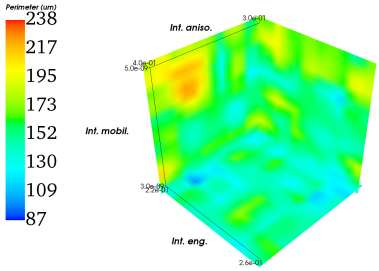
Figure 3.8: 3D contours of dendritic area as a function of thermodynamic parameters, where processing parameters are varied.



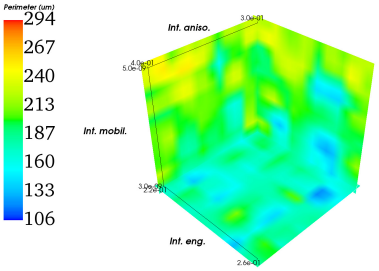
(a) dendritic perimeter as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 915K, respectively.



(b) dendritic perimeter as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 915K, respectively.

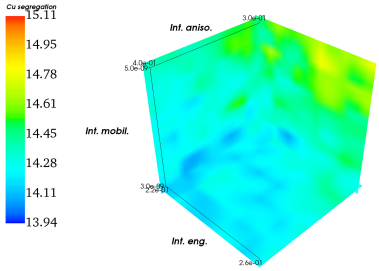


(c) dendritic perimeter as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 920K, respectively.

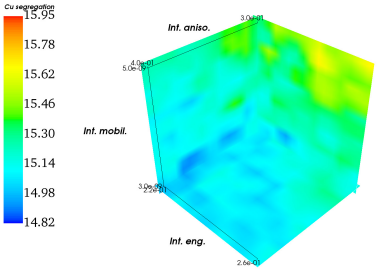


(d) dendritic perimeter as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 920K, respectively.

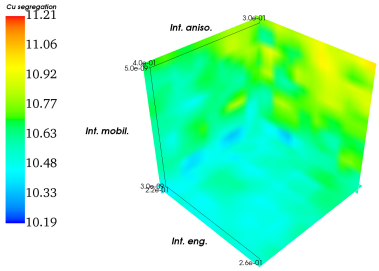
Figure 3.9: 3D contours of dendrite parameter as a function of thermodynamic parameters, where processing parameters are varied.



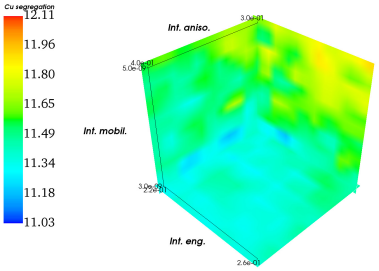
(a) Cu segregation as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 915K, respectively.



(b) Cu segregation as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 915K, respectively.

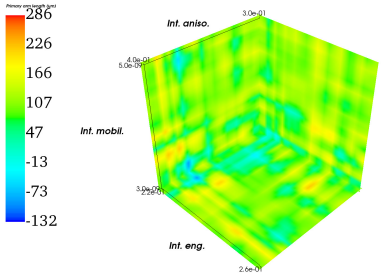


(c) Cu segregation as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 920K, respectively.

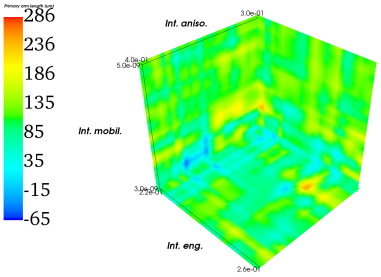


(d) Cu segregation as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 920K, respectively.

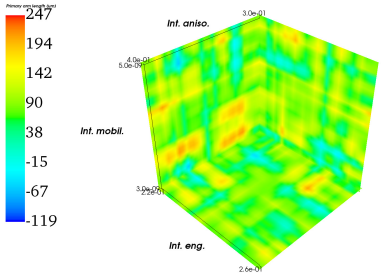
Figure 3.10: 3D contours of Cu segregation as a function of thermodynamic parameters, where processing parameters are varied.



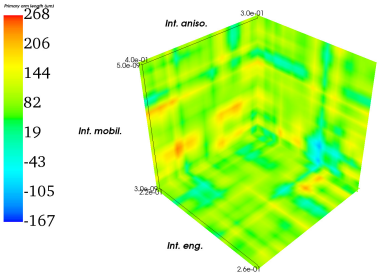
(a) dendritic primary arm length as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 915K, respectively.



(b) dendritic primary arm length as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 915K, respectively.



(c) dendritic primary arm length as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -10K/s and 920K, respectively.



(d) dendritic primary arm length as a function of thermodynamic parameters. Cooling rate $\frac{\partial T}{\partial t}$ and initial temperature T_0 are fixed at -20K/s and 920K, respectively.

Figure 3.11: 3D contours of dendritic primary arm length as a function of thermodynamic parameters, where processing parameters are varied.

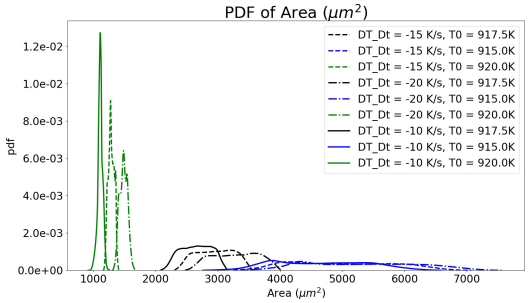
3.5.2 Stochastic collocation: random thermodynamic and deterministic processing parameters

In this section, a UQ analysis is conducted using stochastic collocation method, which integrates SG and PCE, to study the distribution of the QoIs. The processing parameters, T_0 and $\frac{\partial T}{\partial t}$, are set to be deterministic with physical meaning that those variables are controllable in practice. The thermodynamic parameters, σ_0^*, ε^*, and M_ϕ, however, are assumed to be random, due to the assumptions that these parameters are often calibrated by other ICME tools, such as CALPHAD and *ab-initio* calculations. The thermodynamic parameters are assumed to be uniformly distributed between the lower and upper bounds according to Table 3.2.

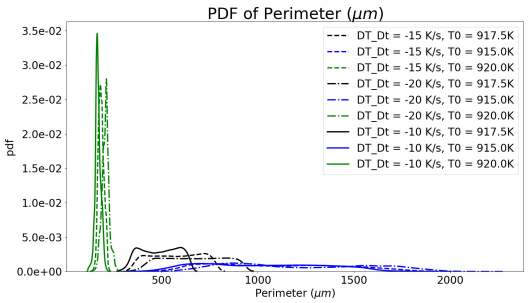
UQToolKit [98, 99] is utilized to calculate PCE coefficients, which are obtained by Galerkin projection in theory. Orthogonal Legendre polynomials of sixth order are used in constructing the PDF of the QoIs, where the thermodynamic parameters are random. A 2-dimensional SG is constructed for processing parameters. A PDF is constructed at each node of the SG. The mean and standard deviation is then calculated for each node. The mean and standard deviation of the QoIs' PDFs are then reconstructed on 2-dimensional processing space using SG formulation.

Figures 3.12a, 3.12b, 3.12c, and 3.12d show the PDF of the dendritic area, dendritic perimeter, Cu segregation, and the dendritic arm length, respectively, where the thermodynamic parameters in Table 3.2, namely σ_0^*, ε^*, and M_ϕ are considered as random variables, which are uniformly distributed between the lower and upper bounds. Several observations are made. First, the dendritic area and perimeter are highly correlated. Second, as shown in Figure 3.12a and Figure 3.12b, the initial temperature T_0 is the dominant factor for the dendritic growth in terms of size, that decreasing the initial temperature T_0 corresponding to larger dendrite size. The cooling rate $\frac{\partial T}{\partial t}$ also affects the dendrite size, but is a less dominant factor. The same observation can be made for the Cu segregation κ, as shown in Figure 3.12c. It is also observed that the standard deviation of the Cu segregation κ only changes mildly, with different processing parameters, as opposed to substantial changes in

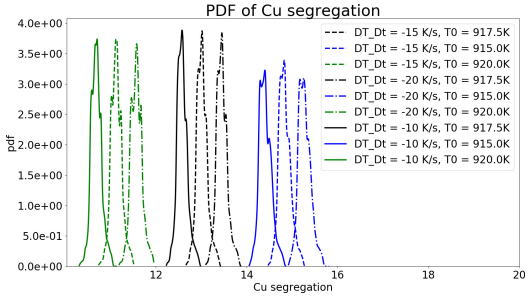
the standard deviations of dendritic area and dendritic perimeter. The dendritic primary arm length is more unpredictable, as shown in Figure 3.12d, where the mean deviates slightly around $100\mu m$. The observations are completely consistent with the previous observation in Section 3.5.1.



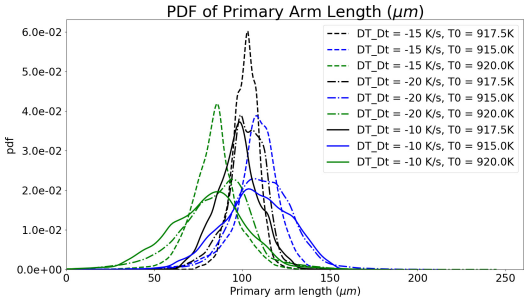
(a) PDF of dendritic area (μm^2).



(b) PDF of dendritic perimeter (μm).



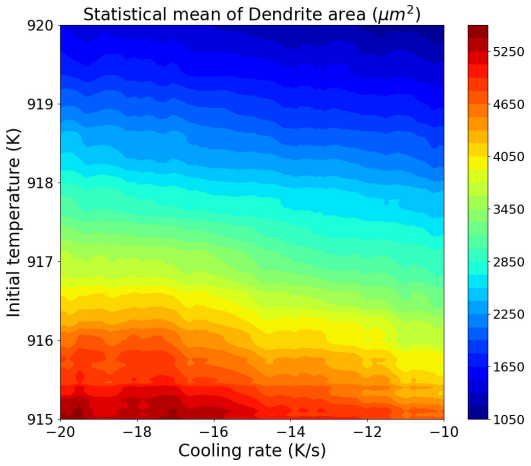
(c) PDF of Cu segregation.



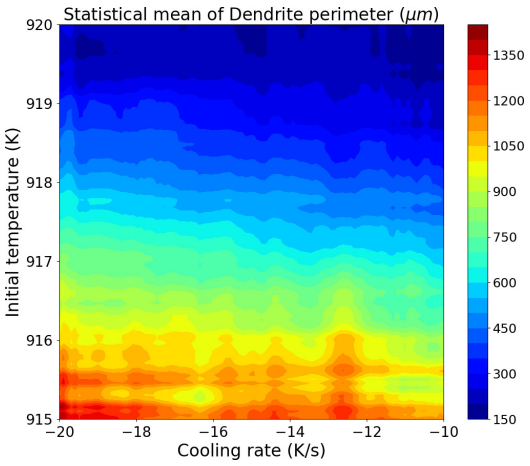
(d) PDF of dendritic primary arm length (μm).

Figure 3.12: Probability density function of QoIs at different processing parameters, where the thermodynamic parameters are distributed uniformly between the lower and upper bounds in Table 3.2.

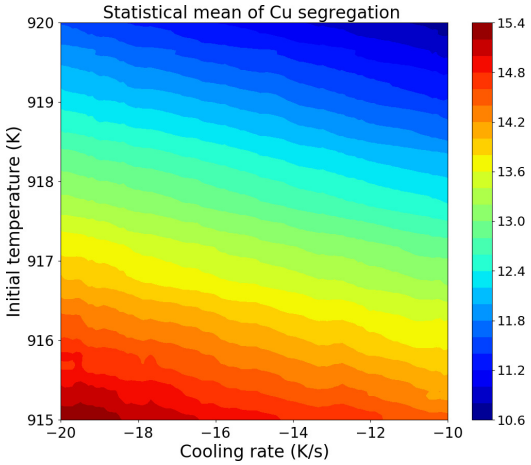
Figure 3.13 and Figure 3.14 show the prediction map of the mean and standard deviation, respectively. The predicted mean is fairly similar, but more noisy compared to the prediction shown in Figure 3.5. Figure 3.14 indicates that there is a weak correlation between the statistical standard deviation and the statistical mean for the first three QoIs, namely the dendritic area, the dendritic perimeter, and the Cu segregation. The dendritic primary arm length appears to be a nonlinear function, and weakly depends on the initial temperature.



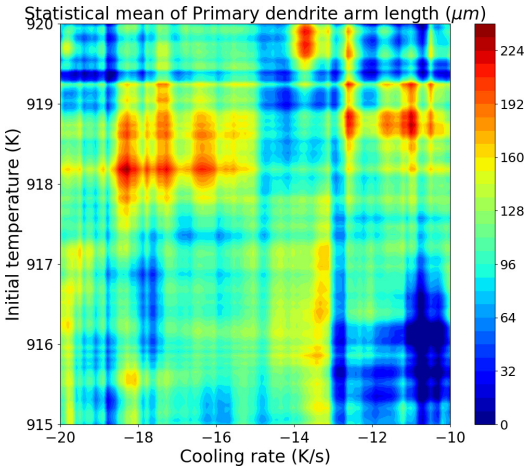
(a) Mean of dendritic area (μm^2).



(b) Mean of dendritic perimeter (μm).

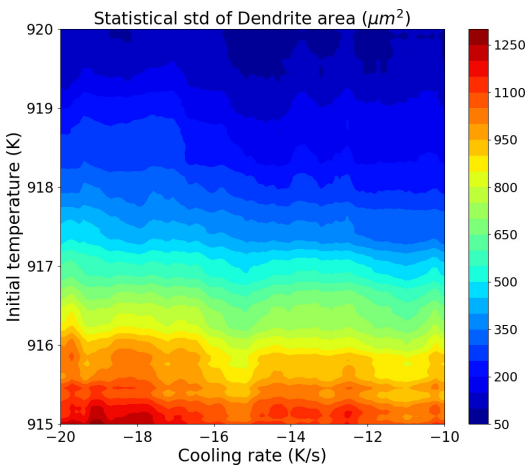


(c) Mean of Cu segregation.

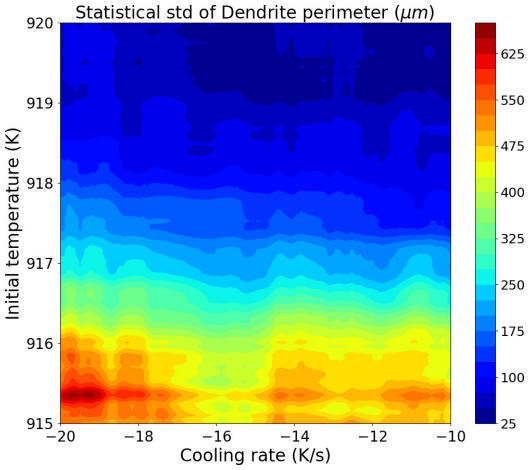


(d) Mean of dendritic primary arm length (μm).

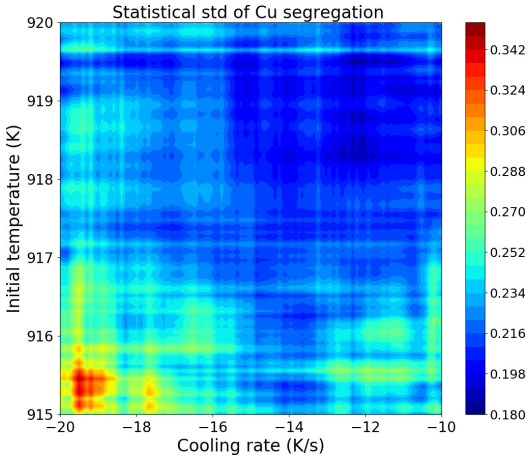
Figure 3.13: Statistical mean of the QoIs, where PDFs are shown in Figure 3.12.



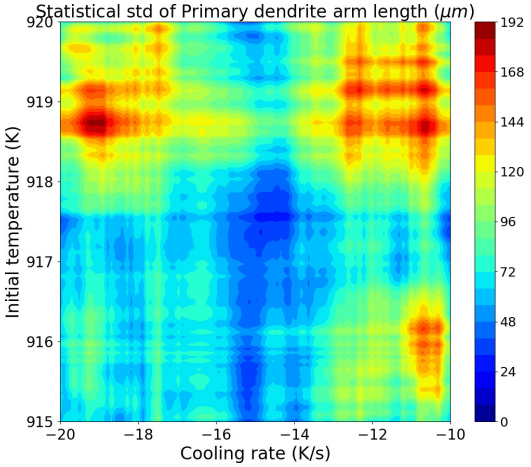
(a) Pdf std of dendritic area (μm^2).



(b) Pdf std of dendritic perimeter (μm).



(c) Pdf std of Cu segregation.



(d) Pdf std of dendritic primary arm length (μm).

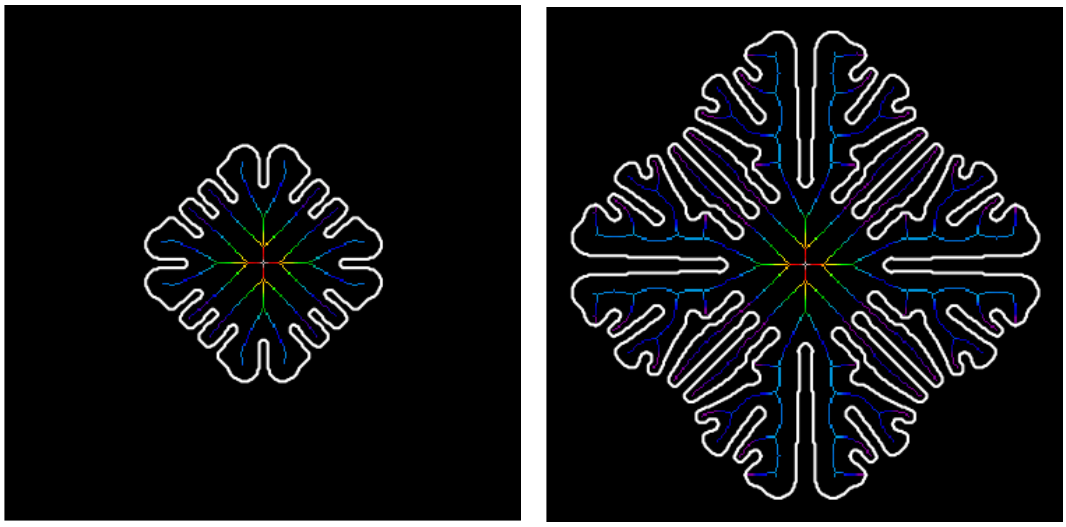
Figure 3.14: Statistical standard deviation of the QoIs, where PDFs are shown in Figure 3.12.

3.6 Discussion

In this UQ study, we investigate the dendrite properties with respect to thermodynamic and processing parameters. While processing parameters can be controlled, the thermodynamic parameters are materials properties and cannot be controlled, but can be quantified with certain experimental errors. Different thermodynamic parameters result in different dendritic morphology and properties.

The variation of dendrite properties with respect to the thermodynamic parameters can be used as a guide for thermodynamic parameters calibration, both qualitatively and quantitatively. Quantitative analysis includes matching experimental and simulated dendritic morphology, rough estimation of dendrite size with respect to solidification time. While dendrite shape can be qualitatively measured, shape description can also be extracted via other image processing techniques. Skeletonization with medial axis method [100] is another example. Figure 3.15 presents the qualitative shape analysis based on skeleltonization technique. The extracted skeleton contains shape information, which can be used for shape matching. From the experimental perspective in materials science, one of the main challenges in shape matching is the inherent randomness of the surface or volume element. The randomness, which is typically referred to as aleatory uncertainty, which in turn, requires the shape analysis to be quantified and modeled as random variables, such as random shape descriptors.

The processing parameters are typically controlled during the manufacturing process, where sensors can be embedded and controllers are activated. However, the controlling process is not always flawless, as unforeseeable problems can be encountered, which leads to imperfect control conditions. Furthermore, even if the controlling conditions are perfect, the processing high-dimensional space needs to be fully explored, in order to understand the property variation with changes in process. Thus, a link of process-structure-property is demonstrated. For example, the Cu segregation κ is investigated as a function of process-



(a) Dendrite skeleton at timestep 3000. (b) Dendrite skeleton at timestep 6000.

Figure 3.15: Skeletonization of the dendrite in Figure 3.2 at different snapshots.

ing parameters. The Cu segregation indicator κ, as shown in Figure 3.5c, increases with lower initial temperature and higher cooling rate. It is also known that high κ is also associated with higher mechanical strength, which is a mechanical property, as θ phase Al_2Cu is promoted. Thus, exploring the microstructure variation with respect to the processing parameters plays an important role in devising an optimal process, which corresponds to an optimal property.

3.7 Summary

In this chapter, a UQ study to investigate the process-structure relationship in Al-4wt%Cu is conducted on PFM using SG to mitigate the effect of dimensionality. Five input parameters are included, where two of them are processing parameters, and the other three are thermodynamic parameters. Image processing techniques are utilized to analyze the dendritic morphology qualitatively. The UQ study is conducted based on SG with high-dimensional interpolation framework. 19313 PFM simulations are performed with different input parameters, where the bounds are chosen *a priori*. It is noted that the summary is drawn within the bound of processing and thermodynamic parameters. The count of secondary

arm does not necessarily correlate with the dendritic area and dendritic perimeter. The dendritic area is shown to correlate with the dendritic perimeter, i.e. dendrite with larger area also has a longer perimeter. The UQ study aims to establish the structure-property relationship in term of dendritic morphology in high-dimensional input space, which includes both thermodynamic and processing parameters. The UQ study also explores the variation of dendrite properties, in which thermodynamic parameters plays a major role. The computational insights aid the thermodynamic parameters calibration and help researchers to understand the variation of dendrite structure with respect to uncertain thermodynamic parameters, which cannot be controlled. Both processing and thermodynamic parameters are shown to affect the dendritic growth, resulting in different microstructure, and thus different materials properties.

CHAPTER 4

OPTIMIZATION UNDER UNCERTAINTY: MIXED-INTEGER BAYESIAN OPTIMIZATION

Bayesian optimization (BO) is a global optimization method that has the potential for design optimization. However, in classical BO algorithm, the variables are considered as continuous. In real-world engineering problems, both continuous and discrete variables are present. In this work, an efficient approach to incorporate discrete variables to BO is proposed. In the proposed constrained mixed-integer BO method, the sample set is decomposed into smaller clusters during sequential sampling, where each cluster corresponds to a unique ordered set of discrete variables, and a Gaussian process regression (GP) metamodel is constructed for each cluster. The model prediction is formed as the Gaussian mixture model, where the weights are computed based on the pair-wise Wasserstein distance between clusters, and gradually converge to an independent GP as the optimization process advances. The definition of neighborhood can be flexibly and manually defined to account for independence between clusters, such as in the case of categorical variables. Theoretical results are provided in concert with two numerical and engineering examples, and two examples for metamaterial developments, including one fractal and one auxetic metamaterials, where the effective properties depends on both the geometry and the bulk material properties.

4.1 Introduction

Designing materials is to identify structures at micro- and nano-scales to achieve the desirable properties. The major process of design is to establish structure-property relationships, based on which design optimization can be performed. Simulation tools at multiple scales (from atomistic to continuum) have been developed to accelerate this process. Nev-

ertheless, the major technical challenges of efficiency and accuracy still exist. The first one is searching in high-dimensional design space to find the global optimum of material compositions and structural configurations. The second one is the uncertainty associated with the high-dimensional structure-property relationships, which are usually constructed as surrogate models or metamodels. Particularly, aleatory uncertainty can be linked to natural randomness of materials (e.g. grain sizes and grain shapes in polycrystalline materials). Epistemic uncertainty is mainly due to approximations and numerical treatments in surrogates and simulation models. Methods of searching globally for optimal and robust solutions are needed.

BO is a metamodel-based methodology to seek for the global optimal solution under uncertainty in the search space with sequential sampling. Compared to other bio-inspired global optimization algorithms, such as ant colony systems, particle swarm, and genetic algorithm (GA), it has the advantage of maintaining the global search history by constructing a metamodel to approximate the objective function. Typically the metamodel is based on the GP method, and actively updated as more samples are collected. However, in current formulation of GP, input variables are restricted to be continuous. In real-world engineering problems, input design variables and parameters can be categorical or discrete. For example, binary variables can be used to enable or disable a design feature. The number of features has integer values. Therefore extending BO method to accommodate discrete variables is an important topic for solving real-world problems.

Another major issue that prohibits the BO and GP framework is its lack of scalability in searching the high-dimensional space when the number of input variables is large. The required number of sample points grows exponentially as $\mathcal{O}(s^d)$ with respect to the dimension of search space d, where s is the number of sampling point for each dimension. The phenomenon is referred to as the curse-of-dimensionality in literature. As a result, the size of the covariance matrix in GP also grows exponentially with respect to the dimensionality, creating the computational bottleneck in computing the inverse of the covariance matrix.

In this paper, a new BO method is proposed for constrained mixed-integer optimization problems to incorporate discrete design variables into the BO algorithm. In the proposed method, the large dataset of samples is decomposed into smaller clusters, where each cluster corresponds to a unique combination of discrete variable values, which is referred to as a discrete tuple. A GP is then constructed within each cluster. During the search and metamodel update processes, the mean and variance predictions are formulated as a Gaussian mixture model, where the weighted average predictions are combined from those of neighboring clusters, based on the pair-wise distance between the main and the neighboring clusters. The neighborhood of each cluster is constructed only once during the initialization.

Because of the decomposition approach, the number of sampling points to construct each cluster is significantly reduced compared to the whole dataset, and the GP thus is faster to construct for each cluster. This approach, however, leads to an undesirable effect of sparsity within each GP cluster. As a result, the posterior variance might be slightly overestimated. To circumvent the sparsity effect of the decomposition approach, a weighted average scheme is adapted to ”borrow” the sampling points from neighboring clusters, where the discrete tuples of the neighbors slightly differ from the discrete tuple of the original cluster. The definition of neighborhood is completely controlled by users, and neighbors can be added or removed accordingly. The unique advantage of the proposed method is that the optimization problem of both continuous and discrete variables and the acceleration of GP for high-dimensional problems are solved simultaneously. Theoretical results are provided and discussed in concert with computational metamaterials design applications.

In the remainder of the paper, Section 4.2 provides a literature review for BO methodology, its extension, such as constrained and mix-integer optimization problems, and its applications. Section 4.3 describes the proposed constrained mixed-integer BO algorithm using Gaussian mixture model, including theoretical analysis of algorithmic complexity as well as lower and upper bounds of the predictions. The methodology is demonstrated

with applications in computational design of metamaterials. Metamaterials are an emerging class of engineered materials that exhibit interesting and desirable macroscopic properties, which can be tailored, because of their engineered geometric structures rather than the material composition. In Section 4.4, the proposed method is verified using an analytical function that is modified based on a discrete version of Rastrigin function, an engineering example of welded beam design, where the discrete variables encode the material selection and design configuration of the beam. In the first engineering example of Section 4.5.1, we focus on designing high-strength and low-weight fractal metamaterials, where the effective material properties, such as effective Young's modulus is obtained using finite element method (FEM). In the second engineering example of Section 4.5.2, the method is demonstrated using an auxetic metamaterials for polymers, where the effective negative Poisson's ratio is optimized. Section 4.6 includes the discussion of the limitations in the proposed approach, and Section 4.7 concludes the paper, respectively.

4.2 Related work

Here, we conduct a literature review on related BO work and its design applications. In Section 4.2.1, the widely used acquisition functions for BO are introduced. The constrained optimization problem in BO is reviewed in section 5.2.2. In Section 4.2.3, the mixed-integer optimization problem in BO and its relate work is discussed. In Section 4.2.4, the applications of GP in design optimization is provided.

4.2.1 Acquisition function

BO is a metamodel-based optimization framework that uses GP as the metamodel. The major difference between BO and GP based optimization is the sampling strategy to construct the metamodel. The significant extension of BO is the implementation of a so-called acquisition function that dictates the location of the next sampling design site. This acquisition function reconciles the trade-off between exploration (navigating to the most uncertain

region) and exploitation (driving the solution to the optimum) in the optimization process.

Given the objective function $y = f(\boldsymbol{x})$, the acquisition function $a(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta)$ depends on previous N observations or samples $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ and GP hyperparameters θ, and must be defined to strike a balance between exploration and exploitation. In exploration, the acquisition function a would lead to the next sampling point in an unknown region where the posterior variance $\sigma^2(\boldsymbol{x})$ is large. In exploitation, the acquisition function a would result in the next sampling point where posterior mean $\mu(\boldsymbol{x})$ is large for a maximization problem (or small for minimization). There are mainly three types of acquisition functions: probability of improvement (PI), expected improvement (EI), and upper confidence bound (UCB). They are defined as follows.

Let $\boldsymbol{x}_{\text{best}} = arg\,max_{\boldsymbol{x}_i} f(\boldsymbol{x}_i)$ be the best sample achieved so far during sequential sampling for a maximization problem, $\phi(\cdot)$ and $\Phi(\cdot)$ be the probability density function and cumulative distribution function of the standard normal distribution respectively. The PI acquisition function [101] is defined as

$$a_{\text{PI}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \Phi(\gamma(\boldsymbol{x})), \tag{4.1}$$

where

$$\gamma(\boldsymbol{x}) = \frac{\mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) - f(\boldsymbol{x}_{\text{best}})}{\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta)}, \tag{4.2}$$

indicates the deviation away from the best sample. The EI acquisition function [102][103] is mathematically expressed as

$$a_{\text{EI}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) \cdot (\gamma(\boldsymbol{x})\Phi(\gamma(\boldsymbol{x})) + \phi(\gamma(\boldsymbol{x})) \tag{4.3}$$

Recently, Srinivas et al. [104][105] proposed a new form of UCB acquisition function,

$$a_{\text{UCB}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) + \kappa\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta), \tag{4.4}$$

where κ is a hyperparameter describing the exploitation-exploration balance.

4.2.2 Constrained BO

Constrained BO is a natural and important extension of the classical BO method. Constrained optimization problems based on engineering model and simulation can be classified as two types: known and unknown constraints. The known constraints, or a priori constraints, are the ones known before the simulation, and thus can be evaluated independently without running simulations. On the other hand, the unknown constraints are the ones that are unpredictable without running the simulation, and thus can be only incorporated once the simulation is over, e.g. no solution because of numerical divergence. Generally speaking, the unknown constraints are more difficult to assess because it involves handling the classification problem, satisfied or violated, with respect to the optimization problem.

Digabel and Wild [106] summarized and provided a systematic classification and taxonomy for constrained optimization problem. Gardner et al. [107] proposed a penalized acquisition function approach to limit the searching space for the next sampling location. Gelbart et al. [108] suggested an entropy search criterion to search for the next sampling point under the formulation of the EI acquisition function. Hernández-Lobato et al. [109] [110] introduced a predictive entropy search and predictive entropy search with constraints, respectively, which maximizes the expected information gained with respect to the global maximum. Rehman and Langelaar [111] modeled constraints as a simple model and incorporated probability of feasibility measure to alternate the EI acquisition function.

4.2.3 Mixed-integer Bayesian optimization

The BO extension to mixed-integer problems is rather limited, partly because mixed-integer problems carry difficulties from both discrete and continuous optimization problems. Another approach is that the discrete optimization can be converted to continuous optimization, using simple rounding operation. The approach is not mathematically rigorous, but is

still widely accepted in practice. Here we review several contributions in term of methodology to incorporate discrete variables.

Davis and Ierapetritou [112] combined a branch-and-bound approach with BO method to solve the mixed-integer optimization problems. Müller et al. [113, 114, 115] introduced three algorithms, which are Surrogate Optimization-Mixed Integer [113], Surrogate Optimization-Integer [114], and Mixed-Integer Surrogate Optimization [115], which differ in the perturbation sampling strategies and utilize GP as the surrogate model, to solve for the mixed-integer nonlinear problems. Hemker et al. [116] compared the performance of a GA, the implicit filtering algorithm, and a branch-and-bound approach formulated on BO algorithm to solve for a set of constrained mix-integer problems in groundwater management.

For mixed-integer extension for GP, van Stein et al. [117] proposed a distributed kriging approach, where the dataset is decomposed for continuous variables using k-mean algorithm, and the optimal weights are computed based on the inverse posterior variance of each cluster. Gramacy et al. [118] [119] [120] developed a treed GP that is naturally extensible to handle discrete variables. In the case of discrete variables, the GP is one-hot encoded by the binary combination of the discrete variables. Storlie et al. [121] developed the Adaptive COmponent Selection Shrinkage Operato method (ACOSSO), originated from Lin and Zhang [122] [123], which uses the smoothing spline ANOVA decomposition to decompose the total variance to multivariate functions. Qian et al. [124] [125] approached the mixed-integer problem from the covariance kernel of GP, proposing the exchange correlation, the multiplicative correlation, and the unrestricted correlation functions to handle discrete variable that is reminiscent of categorical regression. Swiler et al. [126] compared three above methods and concluded that GP with special correlation kernel [124] [125] performs most consistently among the test functions.

4.2.4 GP-based design optimization

GP, also known as kriging, has been widely applied in constructing surrogates or metamodels for design optimization. Simpson et al. [127], Queipo et al. [128], Martins and Lambe [129], Sóbester et al. [130], and Viana et al. [131] provided comprehensive reviews on the use of kriging and other surrogate models for multi-disciplinary design optimization. More recently, Li et al. [132] proposed a kriging metamodel assisted multi-objective GA to solve multi-objective optimization problems. Jang et al. [133] used dynamic kriging to solve a design optimization in fluid-solid interaction. Zhang et al. [134] also used kriging to approximate the pump performance and optimize two objective functions with respect to four design variables. Kim et al. [135] optimized and verified a fluid dynamic bearings simulation using kriging approach. Kim et al. [136] applied multi-fidelity kriging and optimized film-cooling hole arrangement. Liu et al. [137] employed surrogate-based parallel optimization method to reduce the computational time for a computational fluid dynamics problem with six design variables. Song et al. [138] used a gradient-enhanced hierarchical kriging to optimize drag on airfoils at a specified angle of attack. Zhou et al. [139][140] developed a multi-fidelity kriging scheme to approximate the lift coefficient as a function of Mach number and angle of attack in airfoils with computational fluid dynamics analysis.

In the above work, design variables are all continuous. Compared to these GP-based optimization, BO formulation provides a more generic and robust searching procedure.

4.3 Proposed mixed-integer Bayesian optimization

The proposed mixed-integer BO based on distributed GP provides an efficient searching method for large scale design problems, where design variables can be either continuous or discrete. The discrete variables include both categorical and integer variables, regardless of the existence of order relations. Let $\boldsymbol{x} = (\boldsymbol{x}^{(d)}, \boldsymbol{x}^{(c)})$ be the design variables, where $\boldsymbol{x}^{(d)} \in \mathbb{D}$ are discrete variables in n-dimensional space $\mathbb{D}$ and $\boldsymbol{x}^{(c)} \in \mathbb{R}^{m-n}$ are continuous

variables in $(m-n)$-dimensional space $\mathbb{R}^{m-n}$. Together, they form a vector of design variables in the m-dimensional space $\mathcal{X}$. Let $f(\boldsymbol{x})$ be the objective function. The design optimization problem solves the maximization problem

$$\boldsymbol{x}^* = arg\, max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}), \tag{4.5}$$

subject to some inequality constraints

$$g_i(\boldsymbol{x}) \leq 0, i = 1, \cdots, i_c \tag{4.6}$$

where i_c is the number of inequality constraints.

Here the notation for the rest of the paper is as follows. $\mu_l(\boldsymbol{x})$ is used to denote the posterior mean of the l^{th}-cluster at the query point $\boldsymbol{x}$. $\hat{\mu}$ is the prediction formed by Gaussian mixture model of all the clusters. $\bar{\mu}_l$ is the mean of the l^{th}-cluster.

In the proposed mixed-integer BO, the large dataset of observations is decomposed into smaller local clusters, where each cluster is used to construct a local GP. Because the large dataset has been decomposed and the number of data points has reduced, the prediction within each clusters is not as accurate, and can be improved by "borrowing" from neighboring dataset under a weighted average scheme. The large dataset with continuous and discrete variables can be decomposed to finitely many clusters, according to the tuple of discrete variables. In each cluster, the data points share the same discrete variable values. The classical GP approach is then applied to the dataset in each cluster to construct a GP model.

Because of the decomposition scheme, the number of data points within each cluster is reduced, compared to the number of data points of the whole dataset. This leads to a sparser dataset within a cluster, and the posterior variance is enlarged. To improve the prediction, the datasets from neighboring clusters are initially "borrowed" to improve the prediction on the tuple of continuous variables $\boldsymbol{x}^{(c)} \in \mathbb{R}^{m-n}$, where the "borrowed" data points are

gradually eliminated as the optimization process converges via the weight computation algorithm. On the other hand, the sparsity induced by the decomposition scheme reduces the cost of computing the inverse of the covariance matrix. In this weighted average scheme, the weights are computed and penalized based on the pair-wise Wasserstein distance between clusters, as well as the posterior variance of the cluster to obtain a more accurate predictions to aid in the convergence of the optimization process.

4.3.1 Gaussian process

We follow the notation introduced by Shahriari et al. [141] to briefly introduce GP formulation for continuous variables. GP(μ_0, k) is a nonparametric model that is characterized by its prior mean $\mu_0 : \mathcal{X} \mapsto \mathbb{R}$ and its covariance kernel $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Define $f_i = f(\boldsymbol{x}_i)$ and $y_{1:N}$ as the unknown function values and noisy observations, respectively. In the GP formulation, it is assumed that the $\boldsymbol{f} = f_{1:N}$ are jointly Gaussian and $\boldsymbol{y} = y_{1:N}$ are normally distributed given $\boldsymbol{f}$, then the prior distribution induced by the GP can be described as

$$\boldsymbol{f}|\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{K}), \quad \boldsymbol{y}|\boldsymbol{f}, \sigma^2 \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I}), \tag{4.7}$$

where the elements of mean vector and covariance matrix are described by $m_i := \mu_0(\boldsymbol{x}_i)$ and $K_{i,j} := k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

Equation 4.7 describes the prior distribution induced by the GP, where $\boldsymbol{X}$ is the sampling location, and f is the objective function. In the GP formulation, y is the noise-corrupted stochastic output of $f(\boldsymbol{x})$ with the variance of σ^2, at the sampling location $\boldsymbol{X}$. The objective function f is assumed to be a multivariate normal distribution function f with mean $\boldsymbol{m}(x)$ and covariance $\boldsymbol{K}(x)$.

Let N be the number of sampling locations, and $\mathcal{D}_N = \{\boldsymbol{x}_i, y_i\}_{i=1}^N$ be the set of observations. The covariance kernel k is a choice of modeling the correlation between input locations $\boldsymbol{x}_i$. Covariance functions where length-scale parameters can be inferred through maximum likelihood function is known as automatic relevance determination kernels. One

of the most widely used kernels in this kernel family is the squared-exponential kernel,

$$\boldsymbol{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \theta_0^2 \exp\left(-\frac{r^2}{2}\right), \tag{4.8}$$

where $r^2 = (\boldsymbol{x} - \boldsymbol{x'})\boldsymbol{\Gamma}(\boldsymbol{x} - \boldsymbol{x'})$, where $\boldsymbol{\Gamma}$ is a diagonal matrix of $(m - n)$ squared length scale θ_i.

The posterior Gaussian for the sequential BO is characterized by the mean

$$\mu_{N+1}(\boldsymbol{x}) = \mu_0(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^T(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}(\boldsymbol{y} - \boldsymbol{m}), \tag{4.9}$$

and the variance

$$\sigma^2_{N+1}(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}(\boldsymbol{x})^T(\boldsymbol{K} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{k}(\boldsymbol{x}), \tag{4.10}$$

where $\boldsymbol{k}(\boldsymbol{x})$ is the vector of covariance terms between $\boldsymbol{x}$ and $\boldsymbol{x}_{1:N}$.

4.3.2 Clustering and enumeration algorithm

Assuming that the discrete variables are independent of each other, a clustering and enumeration algorithm is devised to automatically decompose the large dataset to smaller clusters based on the discrete tuple and tag a cluster with a unique index from the enumeration scheme. For the case when some discrete variables are dependent on others, the neighborhood can be manually changed to reflect the knowledge. The set of discrete variables for each cluster are represented as a discrete tuple where each element is a positive integer.

For an integer variable where order relation exists, the discrete variable can simply be represented as a positive integer, e.g. $1 \leq 2$. For a categorical variable where order relation does not exist, such as type of cross section (square or circular), colors (red or blue), type of materials (aluminum or copper), configuration settings, positive integers can still be used. The choice of using tuple of positive integers as a general representation does not affect the

clustering and enumeration scheme, but would affect the construction of neighborhood for each clusters, depending on the nature of discrete variables.

Suppose that the input $\mathbf{x} = (\boldsymbol{x}^{(d)}, \boldsymbol{x}^{(c)}) = (x_1, \cdots, x_n, x_{n+1}, \cdots, x_m)$ includes n discrete and $m-n$ continuous variables. If p_i is denoted as the total number of possible values for discrete variable $x_i, 1 \leq i \leq n$, then the number of clusters is $L = \prod_{i=1}^{n} p_i$. Due to the complexity of possible combinations, each cluster is assigned a unique index in such a way that the map between their discrete variables and cluster index is one-to-one. The index is calculated based on the total ordering of tuples. Without loss of generality, assume that each discrete variable x_i is bounded by $1 \leq x_i \leq p_i$, i.e. $x_i \in \{1, \cdots, p_i\}$ for $1 \leq i \leq n$. Then the relation of lexicographical order, denoted as $\prec$, can be defined for a pair of tuples on the set of all tuples as

$$(a_1, \cdots, a_n) \prec (b_1, \cdots, b_n), \tag{4.11}$$

if and only if $\exists k : 1 \leq k \leq n : (\forall j : 1 \leq j < k : a_i = b_i)$ and $a_k < b_k$, and $1 \leq a_i, b_i \leq p_i$ for all i. With the definition of lexicographical order $\prec$, the cluster index l for the tuple $(a_1, \cdots, a_n)$ can now be calculated as

$$l = \sum_{i=1}^{n-1} (a_i - 1) \prod_{j=i+1}^{n} p_j + a_n. \tag{4.12}$$

Because the index of cluster is uniquely defined based on the tuple of discrete variables, the tuple describing the set of discrete variables can be reconstructed using the index of the cluster, with the quotient and remainder algorithm recursively shown in Algorithm 1. It describes how to construct the set of discrete variables from the cluster index l.

Even though the description of the Algorithm 1 is somewhat complicated, its implementation can be adopted from MATLAB function ind2sub(). The Equation 4.12, which is a reverse version of Algorithm 1, is also implemented using sub2ind() MATLAB function.

Algorithm 1 Reconstruct the tuple of discrete variables $(x_1, \cdots, x_n)$ from cluster index l.

Input: cluster index l, tuple $(p_1, \cdots, p_n)$.
Output: tuple $(a_1, \cdots, a_n)$ of discrete variables

for $i \leftarrow 1, n$ **do**
  if $i \neq n$ **then**
    Find quotient q, remainder r: $q \prod_{j=i+1}^{n} p_j + r = l$
    Set $l \leftarrow r$
    Assign discrete variable: $a_i \leftarrow q + 1$
  else
    Assign discrete variable: $a_n \leftarrow r$
  end if
  Exception if $a_i = 0$
end for
for $i \leftarrow n, -1, 1$ **do**
  if $a_i = 0$ **then**
    $a_i \leftarrow p_i$
    $a_{i-1} \leftarrow a_{i-1} - 1$
  end if
end for

4.3.3 Construction of neighborhood

Consider a cluster with index l, with the tuple of discrete variables $(a_1, \cdots, a_n)$, the neighbors of the l-th cluster $\mathcal{B}(l)$ is the collection of clusters that share most of similarity with the original cluster. Intuitively, the neighborhood is constructed based on the belief of whether there exists a relationship between two clusters.

For example, for integer variables, the discrete tuples of the neighboring clusters may differ in one or a few different integer variables compared to that of the original cluster. In the same manner, for categorical variables, the discrete tuples of the neighboring clusters may differ in one or a few categorical variables compared to that of the original cluster. Based on this description, a possible choice to define the neighborhood $\mathcal{B}(l)$ of the l-th cluster can be mathematically expressed as

$$\mathcal{B}(l) \quad = \quad \{(a_1^*, \cdots, a_n^*) \;\Big|\; d\Big((a_1^*, \cdots, a_n^*), (a_1, \cdots, a_n)\Big) \leq d_{\text{th}}\}, \tag{4.13}$$

where $d\Big((a)_{i=1}^n, (a^*)_{i=1}^n\Big)$ is some metric on a discrete topological space $\mathbb{D}$, and d_{th} is a user-

defined threshold. The metric $d(\cdot,\cdot)$ can be any l_p-norm, for example, Manhattan distance (l_1-norm), or a counting metric of how many discrete (integer and categorical) variables are different between two tuples. It is noted that the metric $d(\cdot,\cdot)$ does not have to strictly obey the definition properties of mathematical norm. In the special case that this metric is set to zero, i.e. $d\Big((a)_{i=1}^n,(a^*)_{i=1}^n\Big)=0$, it likely means that all the clusters are considered to be completely independent of each other. The construction of neighborhood only occurs once during the initialization.

Furthermore, we emphasize that the neighboring list can be manually changed to reflect the physics-based knowledge from the users, or manually constructed to reflect the dependency of the discrete variables. In the case of categorical variables where independence is usually observed, one can simply remove the neighboring cluster from the corresponding categorical variable, as the neighborhood can be manually changed during the initialization phase of the optimization process.

It is recommended to define the neighborhood carefully, as the neighborhood definition has an impact on both convergence rate, and whether the optimization would be trapped at local optimum. The safest setting is to assign $d_{\text{th}}=0$, which assumes clusters are completely independent of each other.

4.3.4 Weights computation

The weight of each cluster's prediction is determined by the Wasserstein distance between the Gaussian posterior of the main cluster with that of the neighboring clusters. Combined together, they form a Gaussian mixture model to predict a response at a query point $\boldsymbol{x}$.

Consider a query point $\boldsymbol{x}$ in the l-th cluster, which has the continuous tuple $\boldsymbol{x}^{(c)}=(x_{n+1},\cdots,x_m)$. Denote the neighborhood of the l-th cluster as $B(l)=\{l^*\}$, where the cardinality of $|B(l)|=k$, i.e. there are k neighbors in the l-th cluster neighborhood. Each of the neighboring cluster l^* can form its own prediction $\mathcal{N}(\mu_{l^*},\sigma_{l^*}^2)$ from the continuous tuple, including $\mathcal{N}(\mu_l,\sigma_l^2)$ for l-th cluster. However, the prediction must be adjusted by

accounting for the bias, i.e. $\text{Bias}_{l^*}[\mu_{l^*}] = \mathbb{E}[\mu_{l^*} - \mu_l] = \bar{\mu}_{l^*} - \bar{\mu}_l$, which is the difference between the mean of two clusters, and the variance $\sigma^2_{l^*}$.

The weight w_{l^*} associated with the prediction from the l^* cluster should be larger for smaller bias $(\bar{\mu}_{l^*} - \bar{\mu}_l)$ and smaller posterior variance $\sigma^2_{l^*}$. The necessity of bias correction is explained later in Theorem 8. Wasserstein distance between two univariate Gaussian $\mathcal{N}(\mu_{l^*}, \sigma^2_{l^*})$ and $\mathcal{N}(\mu_l, \sigma^2_l)$ is provided by Givens et al. [142] as

$$W_2\left(\mathcal{N}(\mu_{l^*}, \sigma^2_{l^*}), \mathcal{N}(\mu_l, \sigma^2_l)\right) = \|\mu_l - \mu_{l^*}\|^2 + \left\|\sqrt{\sigma^2_l} - \sqrt{\sigma^2_{l^*}}\right\|^2 \tag{4.14}$$

Here we propose a deterministic way to compute the numerical weights based on the pair-wise Wasserstein distance, which eventually converges to an independent GP as the optimization process advances. It is easy to see that the W_2-distance of the l-th cluster's prediction to itself is zero, as W_2 is a distance. The weights are computed according to an inverse W_2-distance with a term σ^2_l from the l-th cluster, as

$$w_{l^*} \propto \left[\sigma^2_l + W_2\left(\mathcal{N}(\mu_{l^*}, \sigma^2_{l^*}), \mathcal{N}(\mu_l, \sigma^2_l)\right)\right]^{-1}. \tag{4.15}$$

In Equation 4.15, w_{l^*} are computed based on two factors, the W_2-distance, and the σ^2_l prediction of the l-th cluster. As the optimization process advances, the posterior variance approaches zero, i.e. $\sigma^2_l \to 0$, and the weight scheme converges to a single GP prediction of the corresponding l-th cluster.

4.3.5 Prediction using weighted average of k-nearest neighboring clusters

We model the prediction of a query point using a Gaussian mixture distribution, where the weights are computed on the statistical Wasserstein distance. To predict an unknown query point $\boldsymbol{x} = (\boldsymbol{x}_d, \boldsymbol{x}_c) = (x_1, \cdots, x_n, x_{n+1}, \cdots, x_m)$, we first find the cluster in which $\boldsymbol{x}$ belongs to, and its neighboring clusters. Assume that $\boldsymbol{x}$ belongs to the l-th cluster, and there are k-neighboring clusters.

The principle for weight computation is as follows. As the bias increases, the contributed weight of the prediction w_{l^*} from the l^*-th cluster to l-th cluster is reduced to a smaller value. Also, as the bias, i.e. the pair-wise distance between clusters increases, the contributed weights also decreases. The weight vector are normalized at every step, and eventually converges to a single GP prediction with the weight vector of $[0, \cdots, 1, \cdots, 0]$, where the 1 is located as the l-th cluster.

Since $\boldsymbol{x}$ is located within the l-th cluster, the weight from the l-th cluster is the highest, i.e. as $l^* = l$, then $\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*} = \mu_l$, which is the GP prediction for the l-th cluster. The posterior mean of the proposed method is written as

$$\hat{\mu} = \sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left(\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*} \right), \tag{4.16}$$

where the sum is taken over the list of neighboring cluster from the main cluster l^{th}. $\bar{\mu}_l$ and $\bar{\mu}_{l^*}$ denote the mean of the l-th and l^*-th clusters, respectively. w^* denotes the weight corresponding to the l^*-th cluster, computed once the discrete tuple $\boldsymbol{x}^{(d)}$ of the query point $\boldsymbol{x} = (\boldsymbol{x}^{(d)}, \boldsymbol{x}^{(c)})$ is determined. The posterior varianced of the proposed method is written as

$$\hat{\sigma}^2 = \sum_{l^* \in \mathcal{B}(l)} w_{l^*}^2 \sigma_{l^*}^2, \tag{4.17}$$

where $\sigma_{l^*}^2$ denotes the posterior variance associated with the continuous tuple $\boldsymbol{x}^{(c)}$ of the query point $\boldsymbol{x} = (\boldsymbol{x}^{(d)}, \boldsymbol{x}^{(c)})$.

The prediction scheme for mean $\hat{\mu}(\boldsymbol{x})$ and variance $\hat{\sigma}^2(\boldsymbol{x})$ for an arbitrary location $\boldsymbol{x}$ using Gaussian mixture model can be summarized in Algorithm 2.

4.3.6 Constrained acquisition function in mixed-integer Bayesian optimization

The acquisition function is adopted from Gardner et al. [107] for inequality constraints, and further extended to accomodate discrete and continuous variables to solve for the constrained mixed-integer optimization problems.

Algorithm 2 Prediction using weighted average GP from nearest neighboring clusters.

Input: location $\boldsymbol{x} = (x_1, \cdots, x_n, x_{n+1}, \cdots, x_m)$, mean output of each cluster $\bar{\mu}_{(\cdot)}$
Output: Gaussian mixture posterior mean $\hat{\mu}$ and posterior variance σ^2

1: Find cluster index l corresponding to $\boldsymbol{x}^{(d)} = (x_1, \cdots, x_n)$ ▷ locate the l-th cluster
2: Construct a neighborhood $\mathcal{B}(\cdot)$ for each cluster ▷ query $\boldsymbol{x}$ in all neighboring clusters
3: **for** $l^* \in \mathcal{B}(l)$ **do**
4: Compute GP posterior of the l^*-th cluster: $\hat{\mu}_{l^*}, \sigma^2_{l^*}$
5: **end for**
6: Compute weight $w_{l^*} \propto \left[\sigma_l^2 + W_2\left(\mathcal{N}(\mu_{l^*}, \sigma^2_{l^*}), \mathcal{N}(\mu_l, \sigma_l^2)\right)\right]^{-1}$ ▷ pair-wise Wasserstein distance
7: $w_{l^*} \leftarrow \frac{w_{l^*}}{\sum_{l^* \in \mathcal{B}(l)} w_{l^*}}$ ▷ weight normalization
8: $\hat{\mu} \leftarrow \sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left(\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right)$ ▷ Gaussian mixture posterior mean
9: $\hat{\sigma}^2 \leftarrow \sum_{l^* \in \mathcal{B}(l)} w_{l^*}^2 \sigma^2_{l^*}$ ▷ Gaussian mixture posterior variance
10: Update the average mean of the l-th cluster $\bar{\mu}_l$

First, the constraint is checked using an indicator function $\mathcal{I}(\boldsymbol{x})$ for all i_c constrained inequalities as,

$$\mathcal{I}(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \forall 1 \leq i \leq i_c : g_i(\boldsymbol{x}) \leq 0, \\ 0 \text{ if } \exists 1 \leq i \leq i_c : 0 \leq g_i(\boldsymbol{x}). \end{cases} \tag{4.18}$$

The constrained acquisition function can be considered as the product of the classical acquisition function. As a result, the acquisition function is assigned to have zero value for infeasible region. The penalized approach can be implemented directly into the auxiliary optimizer, which is used to maximize the acquisition function in BO.

In distributed GP, an input $\boldsymbol{x}_{\text{next}} = (x_1, \cdots, x_n, x_{n+1}, \cdots, x_m)$ is comprised of both discrete and continuous variables. For each cluster corresponding to a unique set of discrete tuple $(x_1, \cdots, x_n)$, a distinct next sampling point associated with each cluster is located by maximizing the acquisition function on the tuple of continuous variables $(x_{n+1}, \cdots, x_m)$ for each iteration, in the same manner as classical BO. These next sampling points are retained within the respective clusters. However, only the sampling point whose acquisition function achieves the maximal value among all clusters is chosen, and a new sampling point within that cluster is located and updated for the corresponding cluster. The sampling procedure repeats over again until the optimization criterion is met. In other words, the

next sampling point is chosen as

$$\boldsymbol{x}_{\text{next}} = arg\,max_{l^*} arg\,max_{(x_n, x_{n+1}, \cdots, x_m)} \quad a_{l^*}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}, \theta) \cdot \mathcal{I}(\boldsymbol{x}), \tag{4.19}$$

where the l^*-th cluster corresponds to the tuple of discrete variables $(x_1, \cdots, x_n)$, and $\mathcal{I}(\boldsymbol{x})$ is the constraint indicator function.

Equation 4.19, which describes the searching procedure for next sampling point by maximizing the penalized acquisition function, is explained as follows. Two loops are constructed to search for the global sampling point. In the inner loop which searches for the local sampling point within each cluster, the penalized acquisition function is the objective function. Maximizing this penalized acquisition function using an auxiliary optimizer yields the local sampling point for each cluster. In the outer loop, the cluster with maximized acquisition function value is determined. The discrete tuple corresponding to the cluster index, which contains the sampling point with maximum value for the acquisition function, is reconstructed using Algorithm 1. In other words, the sampling location $\boldsymbol{x}$ is decomposed to two part: the inner loop searches for the continuous tuple, whereas the outer loop yields the discrete tuple. Theoretically, once the functional evaluation is over, only the cluster that contains the last sampling location needs to be updated. Practically, all the clusters need to update their corresponding sampling locations $\boldsymbol{x}_{\text{next}}$ after certain number of iterations, in order to avoid trapping in local optimum.

The tuple of continuous variables is found by maximizing the acquisition function, whereas the tuple of discrete variables is assigned according to the cluster index. For the EI and PI acquisition functions, x_{best} is modified to be the best point achieved so far among all clusters. For the UCB acquisition function, no modification is needed, assuming the hyperparameter κ is uniform for all clusters. We note that the balance between exploration and exploitation is preserved locally within each cluster, and thus is also presevered globally for all the clusters.

4.3.7 Theoretical bounds and algorithmic complexity

Here we provide the theoretical lower and upper bounds for predictions and algorithm complexity under the formulation of Gaussian mixture model in Theorem 1 and Theorem 2. Theorem 3 proves that under the formulation of the proposed method, then the largest weight is associated with the main cluster. Theorem 8 explains the necessity of translation in mean prediction so that the expected value of the mean is the same with the expected mean in the main cluster.

Theorem 5. *The Gaussian mixture posterior mean* $\hat{\mu} = \sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left(\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right)$ *is bounded by*

$$\min_{l^*} \left(\hat{\mu}_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right) \leq \hat{\mu} \leq \max_{l^*} \left(\hat{\mu}_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right) \tag{4.20}$$

Proof. The proof for the posterior mean is straightforward, noting that $w_{l^*} \geq 0, \forall l^*$ and $\sum w_{l^*} = 1$. □

Theorem 6. *The Gaussian mixture posterior variance* $\hat{\sigma}^2 = \sum_{l^* \in \mathcal{B}(l)} w_{l^*}^2 \sigma_{l^*}^2$ *is bounded by*

$$(\sum_{l^*} w_{l^*}^2 \sigma_{l^*})^2 \leq \hat{\sigma}^2 \leq \max_{l^*} \sigma_{l^*}^2 \tag{4.21}$$

Proof. For the right-hand side of the variance inequality, observe that

$$\begin{aligned} \hat{\sigma}^2 &= \textstyle\sum_{l^*} w_{l^*}^2 \sigma_{l^*}^2 \leq \sum_{l^*} w_{l^*} \sigma_{l^*}^2 \quad (\text{ because } w_{l^*}^2 \leq w_{l^*}) \\ &\leq (\textstyle\sum_{l^*} w_{l^*}) \max_{l^*} \sigma_{l^*}^2 \leq \max_{l^*} \sigma_{l^*}^2 \quad (\text{ because } \textstyle\sum_{l^*} w_{l^*} = 1) \end{aligned} \tag{4.22}$$

For the left-hand side of the variance inequality, recall the Jensen's inequality: $\rho\left(\frac{\sum_i a_i x_i}{\sum_i a_i}\right) \leq \frac{\sum_i a_i \rho(x_i)}{\sum_i a_i}$, where $\rho(\cdot)$ is a convex function. Substitute $w_{l^*}^2 \to a_i$, $\sigma_{l^*} \to x_i$, and $\rho(x) = x^2$ into the Jensen's inequality, we have

$$\left(\frac{\sum_{l^*} w_{l^*}^2 \sigma}{\sum_{l^*} w_{l^*}^2}\right)^2 \leq \frac{\sum_{l^*} w_{l^*}^2 \sigma_{l^*}^2}{\sum_{l^*} w_{l^*}^2} \quad \text{or} \quad \left(\sum_{l^*} w_{l^*}^2 \sigma_{l^*}\right)^2 \leq \left(\sum_{l^*} w_{l^*}^2\right)\left(\sum_{l^*} w_{l^*}^2 \sigma_{l^*}^2\right) \tag{4.23}$$

Now, note that $\sum_{l^*} w_{l^*}^2 \leq \sum_{l^*} w_{l^*} = 1$. We obtain the left-hand side of the inequality.

□

Theorem 7. *The largest weight is associated with the l-th cluster.*

Proof. Based on the weight formula,

$$w_{l^*} \propto \left[\sigma_l^2 + W_2\left(\mathcal{N}(\mu_{l^*}, \sigma_{l^*}^2), \mathcal{N}(\mu_l, \sigma_l^2)\right)\right]^{-1}, \tag{4.24}$$

it is easy to see that the Wasserstein distance between a cluster with itself is zero. Thus, the right-hand side is always less than σ_l^2, i.e.

$$\sigma_l^2 + W_2\left(\mathcal{N}(\mu_{l^*}, \sigma_{l^*}^2), \mathcal{N}(\mu_l, \sigma_l^2)\right) \geq \sigma_l^2. \tag{4.25}$$

Inversing the last inequality completes the proof. The equality occurs when $l^* = l$. □

Theorem 8. *The expectation of the posterior mean $\hat{\mu} = \sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left(\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right)$ is $\bar{\mu}_l$, i.e. $\mathbb{E}[\hat{\mu}] = \bar{\mu}_l$.*

Proof. Take the expectation of Equation 4.9 for any l-th cluster over the continuous domain, and note that $\mathbb{E}[\boldsymbol{y} - \boldsymbol{m}] = 0$, the mean of the posterior is recovered to the mean of the cluster, i.e.

$$\mathbb{E}[\mu_l(\boldsymbol{x})] = \mu_0(\boldsymbol{x}) = \bar{\mu}_l(\boldsymbol{x}). \tag{4.26}$$

The Equation 4.26 holds for any l-th under the GP formulation. In similar manner, take the expectation of the posterior mean $\hat{\mu}$ from the proposed method over the continuous domain, we arrive at

$$\begin{aligned}
\mathbb{E}[\hat{\mu}] &= \textstyle\sum_{l^* \in \mathcal{B}(l)} w_{l^*} \mathbb{E}\left[\mu_{l^*} + \bar{\mu}_l - \bar{\mu}_{l^*}\right] \\
&= \textstyle\sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left[\mathbb{E}[\mu_{l^*}] + \mathbb{E}[\bar{\mu}_l] - \mathbb{E}[\bar{\mu}_{l^*}]\right] \\
&= \textstyle\sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left[\bar{\mu}_{l^*} + \mathbb{E}[\bar{\mu}_l] - \bar{\mu}_{l^*}\right] \\
&= \textstyle\sum_{l^* \in \mathcal{B}(l)} w_{l^*} \left[\bar{\mu}_l\right] \\
&= \bar{\mu}_l.
\end{aligned} \tag{4.27}$$

The second equality is formed by distributing the expectation operator under linear combination rule. The third equality follows by Equation 4.26 as described above. The fourth equality is formed by cancelling two identical terms $\bar{\mu}_{l^*}$. □

A major problem of GP is its scalability, which originates from the computation of the inverse of correlation matrices. The dataset decomposition has a favorable computational aspect in which the scalability is alleviated. Here we analyze the algorithmic complexity based on the assumption that the size of each cluster is roughly equal. Denote the number of data points for the whole dataset as N, and the number of clusters as k. The computational cost for each operation, such as the inverse computation of correlation matrix, the memory cost to store the correlation matrix, and the operation counts in predicting mean and covariance, are listed in Table 4.1.

Table 4.1: Algorithmic complexity of the decomposition approach

	classical BO	proposed BO
compute inverse correlation matrix	$\mathcal{O}(N^3)$	$k\mathcal{O}\left(\left(\frac{N}{k}\right)^3\right) = \frac{1}{k^2}\mathcal{O}(N^3)$
store correlation matrix	$\mathcal{O}(N^2)$	$k\mathcal{O}\left(\left(\frac{N}{k}\right)^2\right) = \frac{1}{k}\mathcal{O}(N^2)$
mean and variance prediction	$\mathcal{O}(N)$	$k\mathcal{O}\left(\left(\frac{N}{k}\right)\right) = \mathcal{O}(N)$

The decomposition approach clearly has a computational advantage to alleviate the scalability problem in GP, even though it is not completely eliminated.

4.4 Analytical examples

4.4.1 An analytical example of discrete Rastrigin function

In this section, the proposed method is applied on the discrete version of the Rastrigin function, which is an analytical function for testing different optimization methods. To evaluate the effectiveness of the proposed mixed-integer BO method, the optimization performance is compared against GA optimization performance, where the settings for GA are varied to avoid the poor choice of parameters. The MATLAB implementation of the GA is called using `ga()` command. The settings for the GA is as follows. To verify the robustness of

the proposed method, three GA settings are chosen. In the first setting, the population size parameter, the elite count parameter are set to 50, 3, respectively. In the second setting, the population size parameter, the elite count parameter are set to 150, 10, respectively. In the third setting, the population size parameter, the elite count parameter are set to 1500, 10, respectively. Other parameters are left to default value.

Problem statement

The DACE toolbox [143] for classical GPR is extended to include the proposed distributed GPR and Bayesian optimization. In this section, the hybrid Bayesian optimization is demonstrated to find the global minimum on a tiled version of Rastrigin function on 25 clusters, where each cluster corresponds to two discrete variables. The input $\boldsymbol{x} = (i, j, x, y)$ is thus comprised of four variables, in which the first two are discrete, and the last two are continuous, as illustrated in Figure 4.1. The original two-dimensional Rastrigin function is $f(x, y) = 20 + [x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y)]$, where $-5.12 \leq x, y \leq 5.12$. The tiled Rastrigin function is constructed based on a tiled domain of Rastrigin function, where each domain is characterized by a discrete tuple (i, j), and the continuous domain is translated to $-0.75 \leq x_{\text{tiled}}, y_{\text{tiled}} \leq 0.75$ for all clusters. Figure 4.1 illustrates the construction of tiled Rastrigin function, and its relationship with the original Rastrigin function. The relationship between the tiled and original Rastrigin can simply be described by an affine function,

$$x_{\text{orig}} = -3.50 + 1.75(i - 1) + x_{\text{tiled}}; \quad y_{\text{orig}} = -3.50 + 1.75(j - 1) + y_{\text{tiled}}, \tag{4.28}$$

where $-0.75 \leq x_{\text{tiled}}, y_{\text{tiled}} \leq 0.75$.

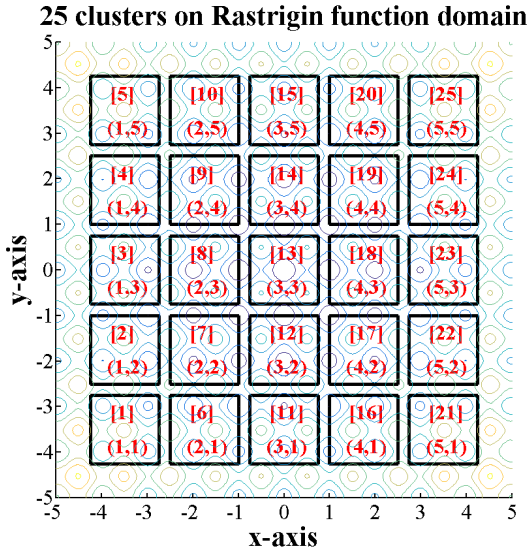


Figure 4.1: Tiled Rastrigin function comprising of 25 clusters, where each cluster correspond to a square of dimension 1.50 $\times$ 1.50 and a tuple (i, j). The cluster index is denoted within the square bracket $[\cdot]$, whereas the tuple is within the parenthesis $(\cdot, \cdot)$ in each square.

Numerical results

In this example, to find the minimum of Rastrigin function, we flip the sign of tiled Rastrigin and used the UCB acquisition function to locate the maximum of the negative tiled Rastrigin function. The covariance matrix adaptation evolution strategy (CMA-ES) [144] method is employed to find the next sampling point within each cluster by locating the point with the maximum acquisition function. The parameters are set as follows: $\kappa = 5$, $d_{\text{penalty}} = 10^{-4}$, $N_{\text{shuffle}} = 15$, where N_{shuffle} is the number of steps which CMA-ES is re-activated with different initial position to search for the next sampling point on each local GPR in order to avoid trapping in the local minima. To construct the initial GPR response surface, 5 random data points are sampled from each cluster.

Because the global minimum of the original Rastrigin function is at $(x = 0, y = 0)$ with the functional evaluation $f(0, 0) = 0$, the hybrid Bayesian optimizer on the tiled Rastrigin function is expected to converge to cluster 13, as illustrated in Figure 4.1. The neighbor

list of cluster 13 includes cluster 8, 12, 13, 14, and 18. Figure 4.2 plots and compares the numerical performance between the proposed mixed integer BO and the GA with three different settings.

Figure 4.2 presents the performance of the proposed method (solid line) with five different settings, and the GA method (dash line) with three different settings. For the proposed mixed integer BO, the threshold distance d_{th} is changed. The proposed mixed integer BO performs best with small d_{th} parameter, which measures the dissimilarity between discrete tuples.

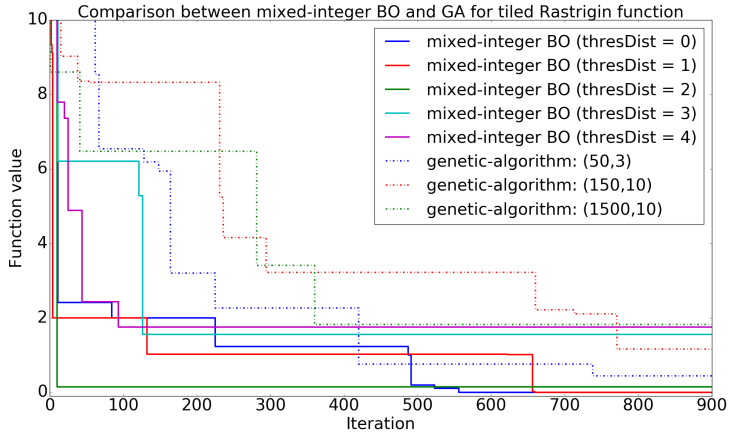


Figure 4.2: Performance comparison between the GA and the proposed mixed-integer BO for the tiled Rastrigin function.

4.4.2 Discrete sphere function

To demonstrate the effectiveness of the proposed mixed-integer BO method in high-dimensional space, a high-dimensional sphere function is modified to leverage the difficulty. The modified high-dimensional sphere function is described as

$$f(x) = \sum_{i=1}^{d} x_i^2 \tag{4.29}$$

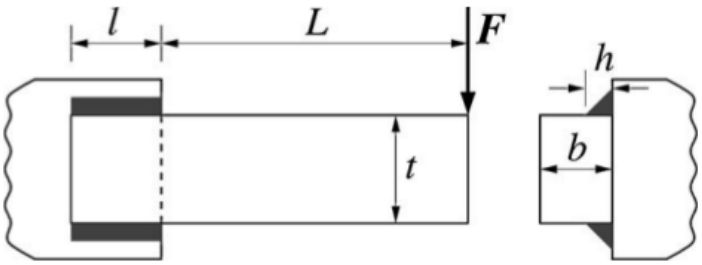


Figure 4.3: Welded beam design problem [148].

4.4.3 Welded beam design problem

To demonstrate the application and verify the proposed method, an analytical engineering model for welded beam design is adapted from Deb and Goyal [145], Gandomi and Yang [146], Rao [147], Datta and Figueira [148], as shown in Figure 4.3, with some slight modifications.

Problem statement

The low-carbon steel (C-1010) beam is welded to a rigid base to support a designated load F. The thickness of the weld h, the length of the welded joint l, the width of the beam t and the thickness of the beam b are the design continuous variables. Two different welding configurations can be used, four-sided welding and two-side welding [145]. The bulk material of the beam can be steel, cast iron, aluminum, or brass, which is associated with different material properties. The stress, deflection, and buckling conditions are derived from Ravindran et al. [149], where the constant parameters are as follows. $L = 14$inch, $\delta_{\max} = 0.25$ inch, and $F = 6,000$lb. The input $\boldsymbol{x}$ is comprised of (w, m, h, l, t, b), where w and m are discrete variables, and h, l, t, b are continuous variables. We note that h, t, b are commonly considered as discrete variables in multiples of 0.0625 in, as well as continuous variables, bounded between lower and upper bounds.

Under this formulation, the objective function is to minimize

$$f(w, m, h, l, t, b) = (1 + C_1)(wt + l)h^2 + C_2 tb(L + l) \tag{4.30}$$

subject to the five inequality constraints:

$$\text{shear stress}(\tau) : g_1 = 0.577\sigma_d - \tau(\boldsymbol{x}) \geq 0 \tag{4.31a}$$

$$\text{bending stress in the beam}(\sigma) : g_2 = \sigma_d - \sigma(\boldsymbol{x}) \geq 0 \tag{4.31b}$$

$$\text{buckling load on the bar}(P_c) : g_3 = b - h \geq 0 \tag{4.31c}$$

$$\text{deflection of the beam} : g_4 = P_c(\boldsymbol{x}) - F \geq 0 \tag{4.31d}$$

$$\text{side constraints} : g_5 = \delta_{\max} - \delta(\boldsymbol{x}) \geq 0 \tag{4.31e}$$

where

$$\sigma(\boldsymbol{x}) = \frac{6FL}{t^2 b}, \delta(\boldsymbol{x}) = \frac{4FL^3}{Et^3 b}, P_c(\boldsymbol{x}) = \frac{4.013tb^3\sqrt{EG}}{6L^2}\left(1 - \frac{t}{4L}\sqrt{\frac{E}{G}}\right) \tag{4.32a}$$

$$\tau = \sqrt{(\tau')^2 + (\tau'')^2 + 2\tau'\tau''\cos\theta}, \tau' = \frac{F}{A}, \tau'' = \frac{F(L + 0.5l)R}{J} \tag{4.32b}$$

$$w = 0 : \begin{cases} A = \sqrt{2}hl \\ J = \sqrt{2}hl\left[\frac{(h+t)^2}{4} + \frac{l^2}{12}\right] \\ R = \frac{1}{2}\sqrt{l^2 + (h+t)^2} \\ \cos\theta = \frac{l}{2R} \end{cases}, \tag{4.32c}$$

$$w = 1 : \begin{cases} A = \sqrt{2}h(t+l) \\ J = \sqrt{2}hl\left[\frac{(h+t)^2}{4} + \frac{l^2}{12}\right] + \sqrt{2}ht\left[\frac{(h+l)^2}{4} + \frac{t^2}{12}\right] \\ R = \max\left\{\frac{1}{2}\sqrt{l^2 + (h+t)^2}, \frac{1}{2}\sqrt{t^2 + (h+l)^2}\right\} \\ \cos\theta = \frac{l}{2R} \end{cases} \tag{4.32d}$$

where w is the binary variable to model the type of weld, $w = 0$ is used for two-sided welding and $w = 1$ is used for four-sided welding. $C_1(m)$, $C_2(m)$, $\sigma_d(m)$, $E(m)$, $G(m)$ are material-dependent parameters [145][146] listed in Table 4.2. The lower and upper bounds of the problem are $0.0625 \leq h \leq 2$, $0.1 \leq l \leq 10$, $2.0 \leq t \leq 20.0$, and $0.0625 \leq b \leq 2.0$

Table 4.2: Material-dependent parameters and constants in the welded beam design problem.

Constants	Description	steel	cast iron	aluminum	brass
C_1	cost per volume of the welded material (\$/in^3)	0.1047	0.0489	0.5235	0.5584
C_2	cost per volume of the bar stock (\$/in^3)	0.0481	0.0224	0.2405	0.2566
σ_d	design normal stress of the bar material (psi)	$30 \cdot 10^3$	$8 \cdot 10^3$	$5 \cdot 10^3$	$8 \cdot 10^3$
E	Young's modulus of bar stock (psi)	$30 \cdot 10^6$	$14 \cdot 10^6$	$10 \cdot 10^6$	$16 \cdot 10^6$
G	shear modulus of bar stock (psi)	$12 \cdot 10^6$	$6 \cdot 10^6$	$4 \cdot 10^6$	$6 \cdot 10^6$

[148].

Numerical results

Here, the input vector is encoded as $\boldsymbol{x} = (w, m, h, l, t, b)$, where $w \in \{0, 1\}$, where $w = 0$ and $w = 1$ correspond to the two-sided and four-sided welding, respectively; $m \in \{1, 2, 3, 4\}$ corresponds to steel, cast iron, aluminum, and brass, respectively.

In this simulation, there are 8 clusters, because there are 2 choices for w and 4 choices for m. The neighborhood $\mathcal{B}(\cdot)$ is considered as universal, i.e. the neighborhood for each cluster includes every clusters, such that they are all aware of others. The bounds for hyper-parameters θ for the GP in each cluster are set as follows. $\underline{\theta} = (0.1, 0.1, 0.1, 0.1)$. $\overline{\theta} = (20.0, 20.0, 20.0, 20.0)$. Every 4 iterations, the sampling point location in each clusters is recomputed again to avoid trapping in local minima. CMA-ES [144] is used as an auxilliary optimizer for maximizing the acquisition function. There are 2 random sampling points in each cluster to initialize the GP construction. The EI acquisition function is used to formulate the acquisition function.

Figure 4.4 shows the convergence plot of the cost function in the welded beam design, where the circle, cross, triangle, and square corresponds to steel, cast iron, aluminum, brass, respectively. The optimal cost value $f(\boldsymbol{x})$ evolves at the iteration 0, 1, 2, 3, 5, and 132, with the value of 20.1995, 5.0605, 3.7949, 3.2436, 1.7420, 1.6297, respectively, with the last one being four-sided welded. Compared to Datta and Figureira [148], where the objective is $f(\boldsymbol{x}) = 1.9553$, our obtained result is smaller $f(\boldsymbol{x}) = 1.6297$, as expected, because in the formulation h, t, and b are continuous variables, in contrast with Datta and

Figureira [148], h, t, b as discrete variables. Furthermore, the convergence occurs relatively fast, as the optimization algorithm exploits the most promising cluster by maximizing the acquisition function. This behavior can be explained by the fact that in this welded beam design example, different materials have significantly different cost objective functional value, which aids the optimization convergence.

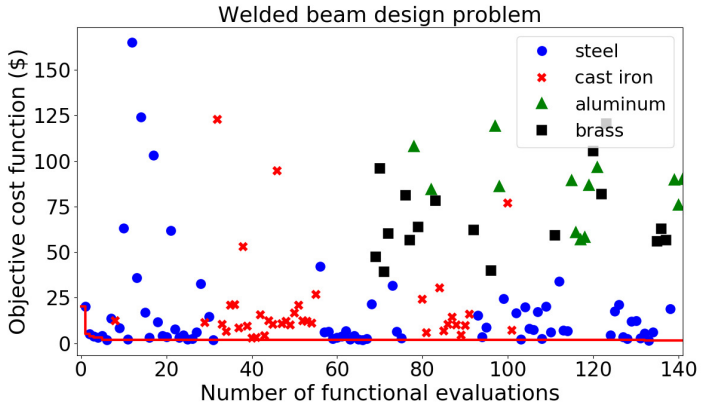


Figure 4.4: Convergence plot of the cost function in the welded beam design, with all clusters are neighbors, showing different combinatorial of discrete and categorical variables are attempted.

To further demonstrate the effectiveness of the proposed method, we compare its numerical with GA. Two versions of the proposed method are used. In the first version, every cluster are considered as independent, leaving no neighbor in the neighborhood, whereas in the second version, all the clusters are considered as neighbors.

The performance comparison is presented in Figure 4.5, showing that both variants of the mixed-integer BO clearly outperforms the GA in the welded beam design problem. The solution obtained from the GA is [0, 1, 0.24920115, 5.30060037, 7.12520087, 0.25345267], where the objective function is evaluated at 2.04016262. On the other hand, from the first variant (none is neighbors) of the proposed method, the solution obtained is [1, 1, 0.16934934, 5.61720010, 4.90884889, 0.27985016], where the objective function is evaluated at 1.68206763. From the second variant (all are neighbors) of the proposed method, the solution obtained is [1, 1, 0.16934934, 5.61720010, 4.90884889, 0.27985016],

where the objective function is evaluated at 1.66457625. The convergence plot of these two variants are very similar to each other. The asymptotical value using the second variant is slightly better than that using the first variant, even though it is hard to conclude which variant performs better. However, we note that as the optimization process advances, the prediction converges to a single GP prediction, and thus asymptotically both variants are similar later on. The proposed mixed integer method clearly outperforms the GA in all settings.

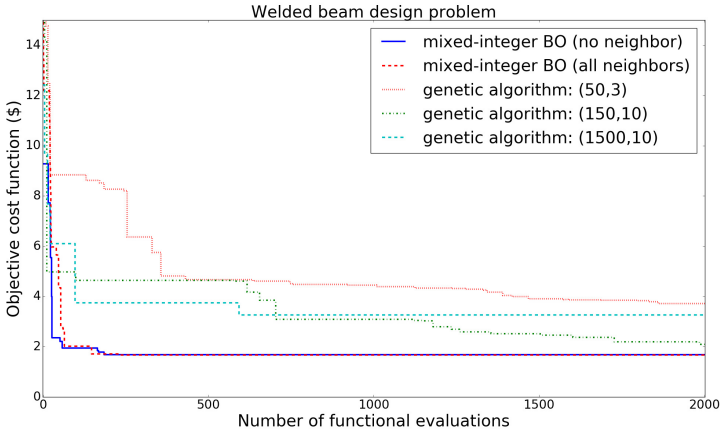


Figure 4.5: Performance comparison between the GA and the proposed mixed-integer BO for the welded beam design.

4.4.4 Pressure vessel design problem

Here, we applied the proposed mixed-integer BO method to solve the pressure vessel design optimization problem. We followed Cagina et al. [150] for the problem formulation. The objective of this problem is to minimize the cost of a storage tank with $3{\cdot}10^3$ psi internal pressure, where the minimum volume is 750 ft^3. The shell is made by joining two hemispheres and forming the longitudinal cylinder with another weld. The design variables are illustrated in Figure 4.7 listed as follows. x_1 is the thickness of the hemisphere. x_2 is the shell thickness. x_3 is the inner radius of the hemisphere. x_4 is the length of the cylinder.

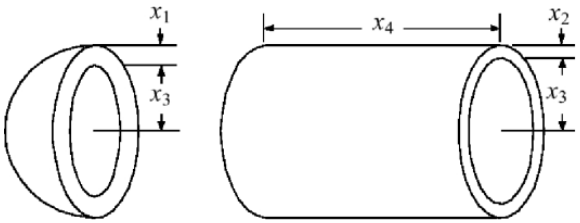


Figure 4.6: Pressure vessel design optimization problem [150].

The objective function that accounts for the cost is

$$f(\boldsymbol{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3, \tag{4.33}$$

where the imposed constraints are

$$g_1(\boldsymbol{x}) = -x_1 + 0.0193x_3 \leq 0, \quad g_2(\boldsymbol{x}) = -x_2 + 0.009541x_3 \leq 0, \tag{4.34a}$$

$$g_3(\boldsymbol{x}) = -\pi x_3^2x_4^2 - \frac{4}{3}x_3^3 + 1296000 \leq 0, \quad g_4(\boldsymbol{x}) = x_4 - 240 \leq 0, \tag{4.34b}$$

and $0.00625 \leq x_1, x_2 \leq 0.61875$, $10.0 \leq x_3$, $x_4 \leq 200.0$. All variables are considered as continuous in this example.

Figure 4.7 shows the performance comparison between the proposed mixed-integer BO and the GA with various settings in terms of number of functional evaluations. Again, the BO clearly shows its advantage in term of convergence speed for continuous variables. The optimal input is [0.193114320, 0.0954997100, 10, 76.2478356], where the corresponding objective functional value is 125.02822748.

4.4.5 Speed reducer design problem

Figure 4.8 shows the design optimization problem of a speed reducer [150]. Seven design variables are described as follows. x_1 is the face width. x_2 is the module of teeth. x_3 is the number of teeth on pinion. x_4 is the length of the first shaft between bearings. x_5 is the length of the second shaft between bearings. x_6 is the diameter of the first shaft.

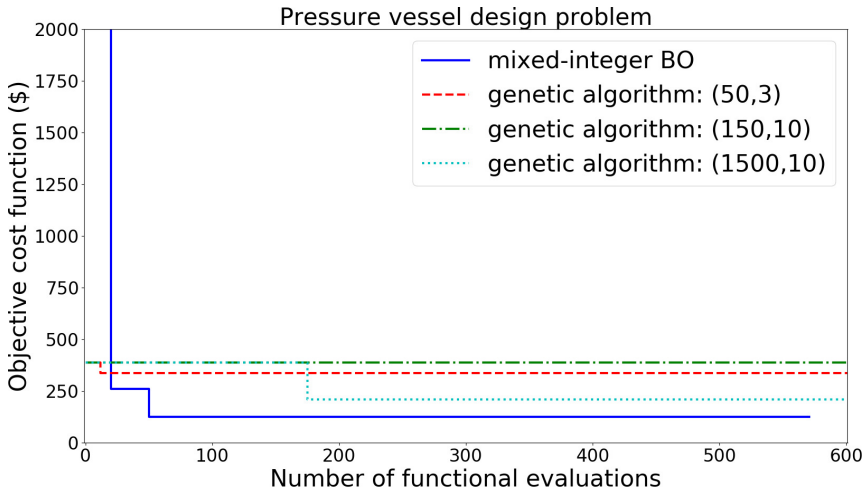


Figure 4.7: Performance comparison between the GA and the proposed mixed-integer BO for the pressure vessel design.

x_7 is the diameter of the second shaft. x_3 is the discrete variable, where the rest of the variables are considered continuous. The problem is 7-dimensional, in which 1 discrete and 6 continuous input space is concerned. With the formulation of the problem, there are 12 local GPs corresponding to 12 discrete values of x_3. In iteration 148, the mixed-

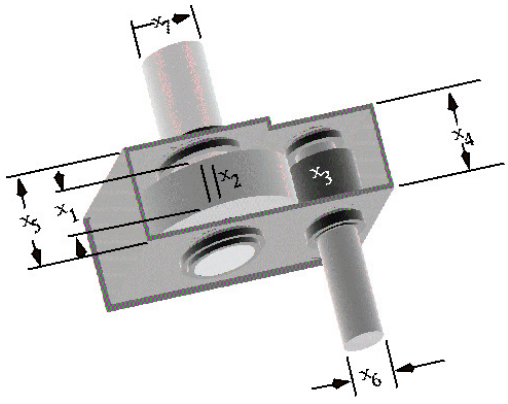


Figure 4.8: Speed reducer design optimization problem [150] from NASA.

integer BO converges to the global minimum of $f(\boldsymbol{x}^*) = 2996.29614837$, where $\boldsymbol{x}^* = [3.50000447, 0.7, 17, 7.30566156, 7.8, 3.35022572, 5.28668406]$. The result is comparable with Cagina et al. [150], where particle swarm optimization is employed, yielding the optimal $f(\boldsymbol{x}^*) = 2996.348165$, where $\boldsymbol{x}^* = 3.5, 0.7, 17, 7.3, 7.8, 3.350214, 5.286683]$.

To evaluate the effect of initial samples, the mixed-integer BO is performed with dif-

ferent number of initial samples. Figure 4.9 shows the convergence plot of the GA and the mixed-integer BO, each with various settings. In terms of the number of functional evaluations, the mixed-integer BO clearly shows the advantages with faster convergence, compared to the GA. The effect of initial samples is also shown in Figure 4.9. It is observed that the proposed mixed-integer BO converges relatively fast after the initial sampling stage. Thus, it may not be necessary to over sample in the initial sampling stage. The balance between exploration and exploitation is well-tuned by the acquisition function, which is GP-UCB [105] in this case.

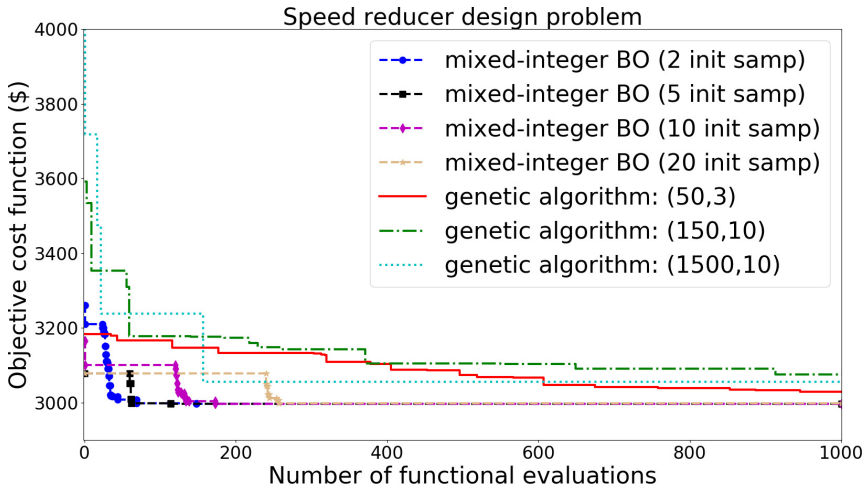


Figure 4.9: Performance comparison between the GA and the proposed mixed-integer BO with different initial samples for the speed reducer design.

4.4.6 High-dimensional discrete sphere function

To evaluate the performance of the proposed mixed-integer BO on high-dimensional space, a discrete sphere function with 5-dimensional discrete variables and 50-dimensional and 100-dimensional continuous variables, resulting in (50+5)D and (100+5)D analytical functions are used to benchmark. The discrete sphere function is described mathematically as

$$f(\boldsymbol{x}^{(d)}, \boldsymbol{x}^{(c)}) = f(x_1, \cdots, x_n, x_{n+1}, \cdots, x_m) = \prod_{i=1}^{n} |x_i| \left(\sum_{j=n+1}^{m} x_j^2 \right) \tag{4.35}$$

where $1 \leq x_i \leq 2, 1 \leq i \leq n$, are integer variables and $-5.12 \leq x_j \leq 5.12, n+1 \leq j \leq m$, are $m-n$ continuous variables. Again, GA is used to compare against the proposed mixed-integer BO method. The global optimal of this function is $f(\boldsymbol{x}^*) = 0$, where $\boldsymbol{x}^* = [1, 1, 1, 1, 1, 0, \ldots, 0]$.

Figure 4.10 shows the convergence plot of the proposed mixed-integer BO with different number of initial samples and GA with different settings for the (100+5)D discrete spherical function, where 5 variables are discrete and 100 variables are continuous. As seen in Figure 4.10, the proposed mixed-integer BO quickly identifies the discrete tuple $(1, 1, 1, 1, 1)$ that correspond minimal response, with respect to the discrete tuple. The rest of the convergence plot focus on the optimization of the continuous variables.

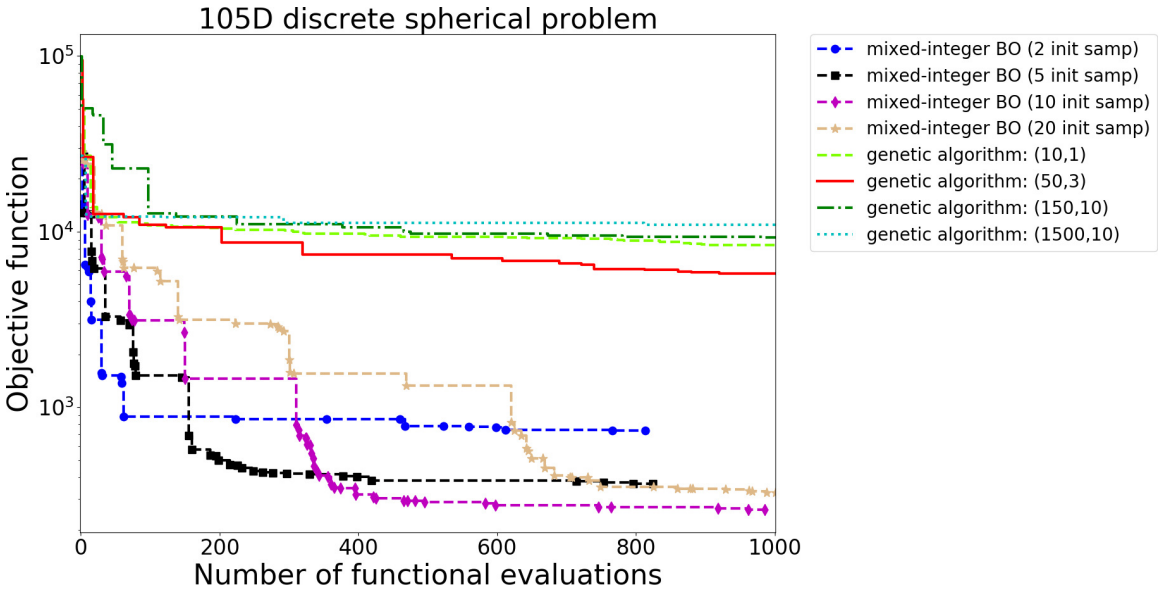


Figure 4.10: Performance comparison between the GA and the proposed mixed-integer BO with different initial samples for (100+5)D discrete spherical function.

4.5 Engineering examples: Metamaterials

In this section, we demonstrate the applicability of the proposed method by examples of a new class of materials, so-called metamaterials, in which properties can be tailored depending on the geometric design of the structures. In Section 4.5.1, a mechanical metamaterial

is considered, where the objective is to design a low-weight and high-strength unit cell. In Section 4.5.2, an auxetic metamaterial unit cell is considered. The proposed BO method is applied to minimize the negative Poisson's ratio.

4.5.1 An example of designing high-strength low-weight fractal metamaterials

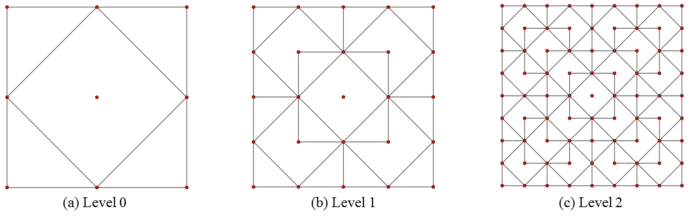
Motivated by the recent experimental work of Meza et al. [151] in designing high-strength and low-weight metamaterials at nano-scale for ceramic systems, which shows the effective mechanical strength can be enhanced by hierarchical structure, we demonstrate the proposed methodology in searching for high-strength and low-weight metamaterials for multiple classes of materials. Particularly, our metamaterials are constructed with fractal geometry. Fractal geometry has the special property of self-similarity at different length scales. A parametric design and optimization approach for fractal metamaterials is demonstrated here. In this example, the goal is to maximize the effective strength of the structure. The effective strength is defined as the ratio between the effective Young modulus and the volume of material with the assumption of homogenized material for the bulk properties. The material selection, including Ashby chart, is formulated as an inequality constraint to limit the searching space of materials.

Parametric design of fractal truss structures

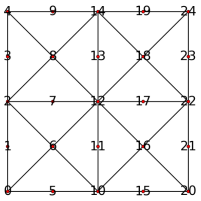
Mathematically, fractals can be constructed iteratively using the so-called iterated function systems (IFSs). An IFS is a finite set of contraction mappings $\{f_i\}_{i=1}^N$ on a complete metric space X [152]. Starting from an initial set $\mathcal{P}_0$, the fractal can be constructed iteratively as $\mathcal{P}_{k+1} = \cup_{i=1}^N f_i(\mathcal{P}_k)$. Geometrically, the IFSs f_i can be expressed in terms of rotation, translation, scaling, and other set topological operations, such as complement, union or intersect.

In this example, the fractal truss structures are constructed from the 2D profiles shown in Figure 4.11a. They are based on the square shape, even though in principle they can be

constructed from any arbitrary polygon such as triangle and hexagon. Figure 4.11a presents the first three levels of IFS construction. The IFSs are inspired by the projection of Keplerian 3D fractals onto its corresponding 2D plane. Here, the IFS operators include the translation matrix $T = \text{diag}\{\pm d/2, \pm d/2, 1\}$ and the scaling matrix $S = \text{diag}\{1/2, 1/2, 1\}$. The rotation is not considered. Physically, the first four IFSs simply scale the design of previous fractal level by $1/2$, and translate them to the northwest, northeast, southwest, and southeast, respectively. The fifth IFS scales the design of previous fractal level by one half, and deletes other features that overlaps within the region.



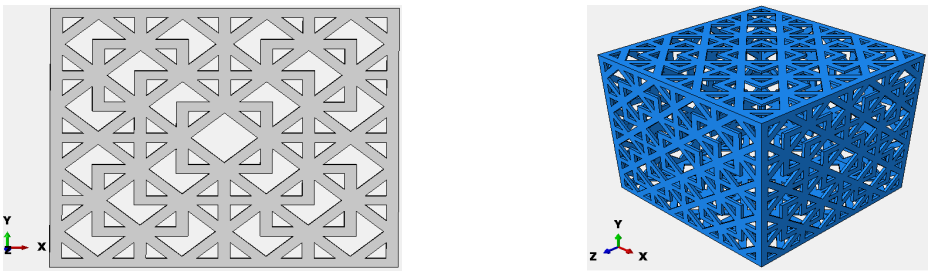
(a) Iterated function systems of truss designs on unit square.



(b) Truss options on fractal level 0 unit square.

Figure 4.11: Truss design paramaters on the unit square.

Figure 4.11b illustrates the square basis with three design options: (1) diagonal truss, (2) inner square truss, and (3) perpendicular truss. The diagonal truss option enables edges connecting nodes 4, 8, 12, 16, 20 and nodes 0, 6, 12, 18, 24. The inner square truss option enables edges connecting nodes 2, 6, 10, 15, 22, 18, 14, 8. The perpendicular truss option enables edges connecting nodes 2, 7, 12, 17, 22 and nodes 10, 11, 12, 13, 14. In the example of Figure 4.11a, only the inner square truss option is enabled. In the construction process, the options are enabled by setting the truss control parameters to 0 or 1, respectively. The fundamental adjacency matrix of fractal level 0 is built to indicate whether a pair of nodes are connected. With the design of level 0 unit cell, the IFSs are applied recursively to create the more complicated geometry at the desired level. Once the profile is constructed, additional offset operations are applied to generate thickness of the 2D truss elements, then extrusion depth for a full 3D structure. Figure 4.12a shows a complete 2D fractal face.



(a) 2D fractal face (b) The unit cube

Figure 4.12: Design of fractal unit cube. (a) The 2D fractal profile with a fractal level of 2 and only inner square truss option enabled. (b) The unit cube is composed of six identical fractal faces, and each face is designed by truss options, thickness, and extrusion depth

With the square face defined, a complete 3D fractal unit cell is built with six of the faces, as shown in Figure 4.12b.

Constitutive material model and the finite element analysis

A general anisotropic material has 21 independent elastic constants to describe the stress-strain (σ-ε) relationship. To simplify the materials constitutive model, we assume isotropic and linear elastic materials behavior at small strain regime, where σ-ε relationship for bulk material properties can be obtained via Young's modulus E and Poisson's ratio ν, i.e.

$$\sigma_{ij} = \frac{E}{1+\nu}\left(\varepsilon_{ij} + \frac{\nu}{1-2\nu}\varepsilon_{kk}\delta_{ij}\right), \tag{4.36}$$

where i, j can be either x, y, or z, and δ_{ij} is the Kronecker delta of i and j. The material properties E and ν, as well as materials ρ, are taken as inputs to describe the linear elastic regime in the FEM simulation to obtain stress.

In simulations, we are concerned with an uniaxial compression. Therefore, to simplify the terminology, we refer to the component of effective stiffness tensor in the loading direction as effective Young's modulus. It is noteworthy that the effective stiffness tensor of the designed fractal truss structure is not the same with the bulk material stiffness tensor. Two displacement boundary conditions are imposed on the unit cube. One is the fixed boundary

condition for both translation and rotation, and the other is the constant displacement on the opposite side of the cube. The stress is obtained by taking the maximum nodal stress in the active direction. The effective Young's modulus is calculated as the ratio of the maximal nodal stress σ_{33} at the designated engineering strain $\varepsilon = 0.01$. The quadratic tetrahedral element (C3D10 in ABAQUS) is utilized for the FEM simulation. The total number of elements is between 5,000 and 10,000. The exact number varies with respect to the finite element simulation. The size of the cube is around 1mm (10^{-3}m).

The dimension of the design space is 9, in which 4 discrete variables and 5 continuous variables are combined to create an input $\boldsymbol{x} = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9)$. The discrete variables include fractal level, the diagonal, inner square, and perpendicular truss options. The fractal level x_1 is an integer of either 0, 1, or 2, whereas each of the truss options x_2, x_3, x_4 is a binary variable from design space, taking value of 0 or 1. The continuous variables include thickness $x_5 = t$ of the truss, the extrusion depth $x_6 = et$ of the unit face, the materials bulk density $x_7 = \rho$, bulk elastic Young's modulus $x_8 = E$, and bulk Poisson's ratio $x_9 = \nu$. Three constraints are imposed as follows. Thickness and extrusion depth are limited to a constant that is related to the fractal level to preserve the fractal geometry of the structure. The higher the fractal level is, the smaller is the constant. Similarly, the material bulk density, Young's modulus, and Poisson's ratio are bounded within a physical limit, where values are taken from Table 3.1 of Bower [153] for woods, copper, tungsten carbide, silica glass, and alloys.

As a result, the imposed constraints are

$$\underline{T} \leq x_5 \leq \overline{T}, \quad x_6 \geq \overline{T}, \tag{4.37a}$$

$$x_5 \leq 7 \cdot x_6, \quad x_6 \leq 7 \cdot x_5, \tag{4.37b}$$

where $\underline{T} = 10^{-6}$ is the threshold for manufacturability $\overline{T}$ is the threshold for the truss

thickness and computed as

$$\overline{T} = \begin{cases} \frac{1}{2 \cdot 2^{x_1+1}}, & \text{if } x_3 = x_4 = 2, \\ \frac{1}{2 \cdot 2^{x_1}}, & \text{otherwise.} \end{cases} \tag{4.38}$$

We expect the simulations to converge on the high-strength and low-density type of materials. However, Ashby chart indicates a high correlation between compressive strength and density among all types of materials. To circumvent this problem, another constraint is introduced to limit the search region, based on the upper bound of longitudinal wave speed as $\sqrt{E/\rho} = \sqrt{x_8/x_7} \leq 10^{4.25}$ m/s.

Simulation and results

Figure 4.13 shows an example of von Mises stress during the uniaxial compression of the architected metamaterial cell, as described in Section 4.5.1. In the simulation settings and its post-process, only σ_{zz} is concerned.

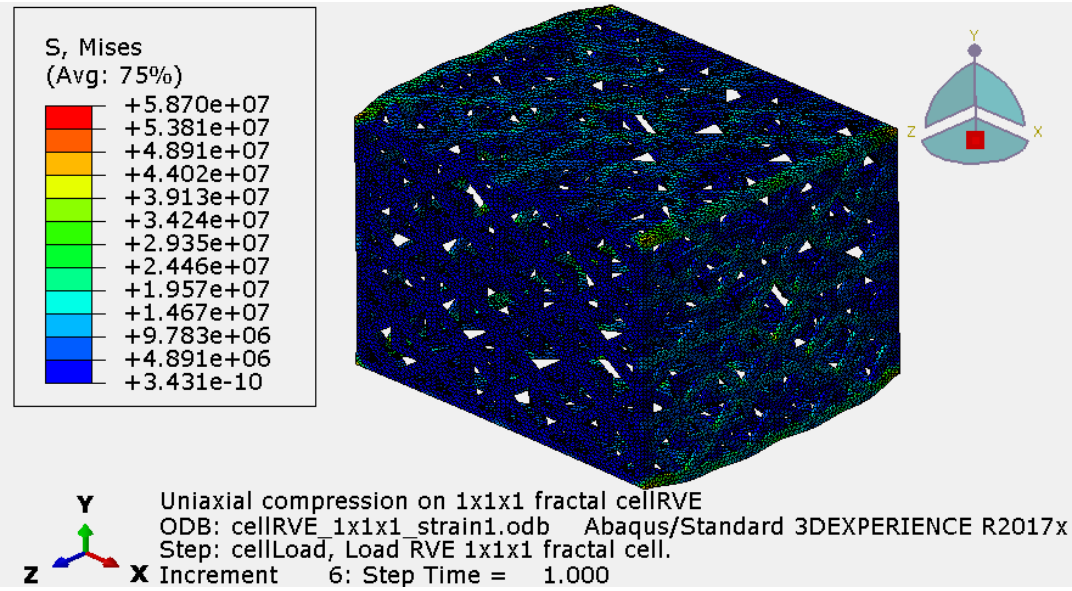


Figure 4.13: An example of von Mises stress of the structure under loading condition.

The lower bounds of continuous variables $(x_5, x_6, x_7, x_8, x_9)$ are $(2 \cdot 10^{-6}, 2 \cdot 10^{-6}, 0.4 \cdot$

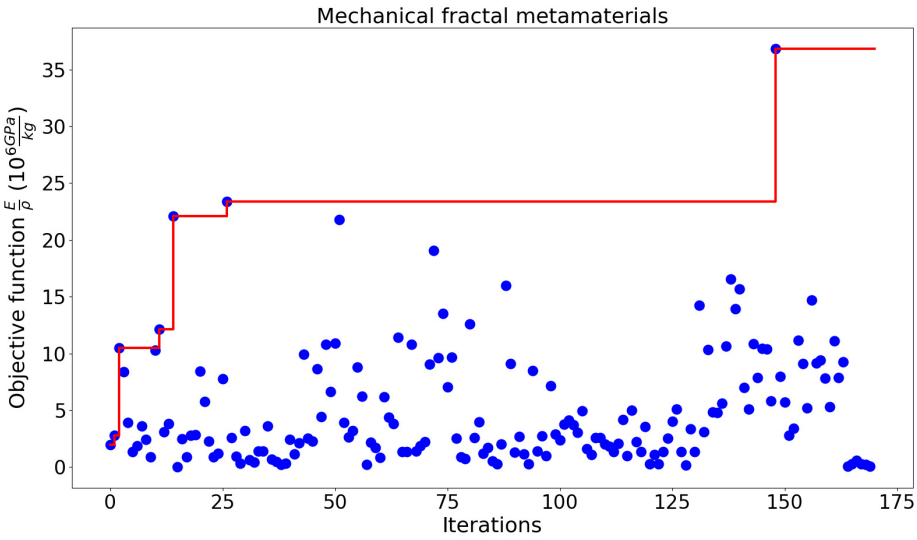


Figure 4.14: Convergence plot of the objective function, which is the ratio between the effective Young's modulus and the weight of the cell, i.e. E_{eff}/m.

$10^{+3}, 9 \cdot 10^{+9}, 0.16)$. The lower bounds of x_7, x_8, x_9 correspond to the density of wood, bulk Young's modulus of wood, and Poisson's ratio of silica glass, respectively. The upper bounds of continuous variables $(x_5, x_6, x_7, x_8, x_9)$ are $(0.5 \cdot 10^{-3}, 0.5 \cdot 10^{-3}, 8.9 \cdot 10^{+3}, 650 \cdot 10^{+9}, 0.35)$. The upper bounds of x_7, x_8, x_9 correspond to the density of copper, bulk Young's modulus of tungsten carbide, and Poisson's ratio of a general alloy, respectively.

To initialize the optimization process, two random inputs are sampled to construct the GP model for each cluster. The number of clusters in this example is $2 \times 2 \times 2 \times 3 = 24$. The EI acquisition is used to locate the next sampling location $\boldsymbol{x}$. The CMA-ES [144] is used as an auxiliary optimizer to maximize the penalized acquisition function. The optimization process is carried out for 170 iterations, as shown in Figure 4.14. At iteration 0, 1, 2, 11, 14, 26, 148, better objective function values of 1.9723, 2.7827, 10.4725, 12.1207, 22.1071, 23.3766, 36.8316 $\cdot 10^6$GPa/kg, are identified, respectively. The relatively fast convergence plot demonstrates the effectiveness of the proposed BO method for the mix-integer optimization problems. Due to the expensive computational cost of the FEM simulation, the number of iterations is limited to 200.

4.5.2 Design optimization of fractal auxetic metamaterials

In the second example, we study the auxetic metamaterial with application in flexible and stretchable devices. Inspired by the experimental work of Cho et al. [154] in designing auxetic metamaterials using fractal cut, and its subsequent numerical and experimental work by Tang et al. [155] in developing shape-programmable materials, we use auxetic metamaterials to demonstrate the proposed BO methodology. The goal of this example is to minimize the effective Poisson's ratio, which is negative and evaluated through a FEM simulation.

Parametric design of auxetic metamaterials

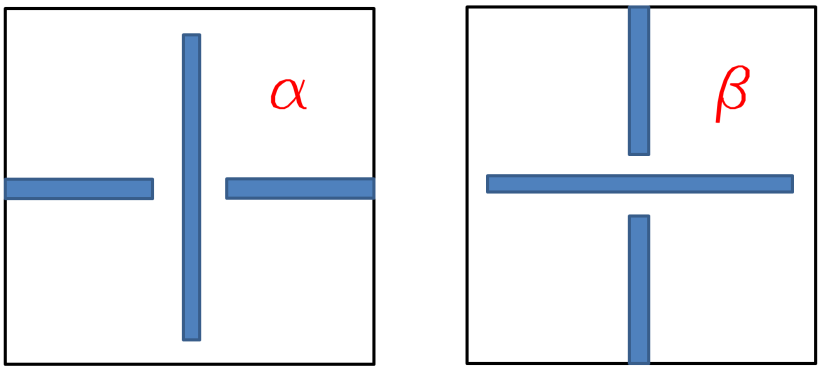


Figure 4.15: Cut motif α and β in designing auxetic metamaterials by fractal cuts.

Here, a parametric design of the unit cell, where the fractal level is fixed at 2, is devised. The cut motif α and β for one level of the auxetic cell is shown in Figure 4.15. Basically, this cut motif controls the free rotational hinges of the architected structure, such that the deformation energy dissipates through rotational motion, rather than translational motion. The principle of cut design is based on the connectivity of the rotating units, where the connectivity depends on the cut patterns, which in turn determine the maximum stretchability of the designed specimen. For further details about the fractal cut and its rotating mechanisms, readers are referred to the work of Cho et al. [154] and Tang et al. [155]. To create a fractal cut, a simple IFS is imposed on the cut to create subsequent level, with the

scaling ratio of $1/2$, and is then translated to four corners.

To tailor the negative Poisson's ratio, the shape of the cut is modeled as splines, where the coordinates of the control points are considered as inputs. The choice of α and β cut is formulated using discrete variables. The dimension of this problem is 18, in which 2 discrete and 16 continuous variables are used. The parametric input $\boldsymbol{x}$ includes x_1, x_2 as discrete variables, which takes value of either 1 (α-motif) or 2 (β-motif) for level 1 and level 2 cuts, respectively. The next 4 continuous variables x_3, x_4, x_5, x_6 is used to describe the shape of the large center cut of level 1. The next 4 continuous variables x_7, x_8, x_9, x_{10} is used to describe the shape of two small side cuts of level 1. In the same manner, the next 8 continuous variables are used to model the large center cut and two small side cuts of level 2. Figure 4.16 shows an example of the parametric design implementation of the designed auxetic metamaterials in the ABAQUS environment. The solid dots present the control points of the cut. (Color is available on the electronic version. The blue solid dots denote the level 1 control points, whereas the red solid dots denote the level 2 control points.)

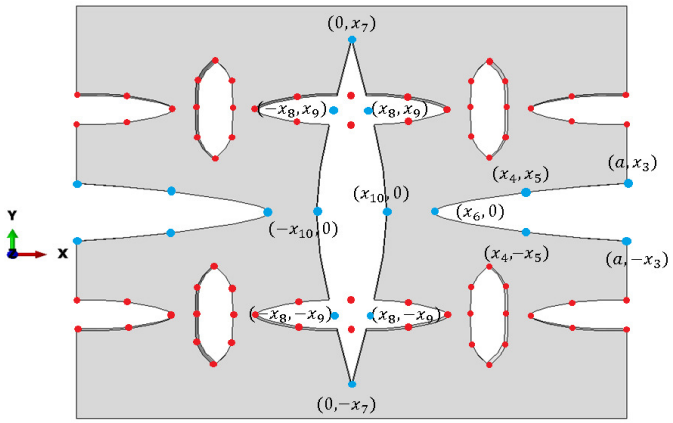


Figure 4.16: An implemented example of auxetic metamaterials by fractal cuts. The solid dots present the control points of the cut. (Color is available on the electronic version. Blue dots correspond to level 1, whereas red dots correspond to level 2.)

Constitutive material model and the finite element analysis

The study of Tang et al. [155] has demonstrated that the effective Poisson's ratio ν_{eff} is indeed a function of strain ε. In this work, we assume that the base material is natural rubber reinforced by carbon-black. Mooney-Rivlin constitutive model is used to describe the hyperelastic material behavior, where the suitable energy function W is expressed as

$$W = C_{10}(\overline{I}_1 - 3) + C_{01}(\overline{I}_2 - 3) + \frac{1}{D_1}(J-1)^2, \tag{4.39}$$

where J is the elastic volume ratio, I_1, I_2, I_3 are the three invariants of Green deformation tensor defined in term of principal stretch ratios $\lambda_1, \lambda_2, \lambda_3$, i.e.

$$I_1 = \sum_{i=1}^{3} \lambda_i^2, \quad I_2 = \sum_{i,j=1;i\neq j} \lambda_i \lambda_j, \quad I_3 = \prod_{i=1}^{3} \lambda_i, \tag{4.40}$$

and $\overline{I}_1 = I_1 J^{-2/3}$, $\overline{I}_2 = I_2 J^{-4/3}$. The materials parameter is adopted from Shahzad et al. [156], where $C_{10} = 0.3339$MPa, $C_{01} = -3.37 \cdot 10^{-}4$, and $D_1 = 1.5828 \cdot 10^{-3}$.

The initial size of the square is 20 cm $\times$ 20 cm, and the thickness of the specimen is 1mm. The specimen is then deformed in a uniaxial tension configuration in y-direction, where the displacement is fixed at 10 cm in one direction. The configuration for the simulation is plane-strain configuration, where displacement in the extrusion direction (z-direction) is fixed as zero.

In the deformed configuration, we extract the displacement in x-direction to infer the engineering transverse strain, and compute the effective Poisson's ratio as the ratio between transverse and longitudinal engineering strain.

The element used in this FEM simulation is the eight-node brick element (C3D8R, C3D6, and C3D4). The FEM is developed using ABAQUS environment. The number of elements for each simulation is approximately 5,000.

In this example, several constraints are imposed on the design variables, which are

$$x_5 \leq 0.010 - \underline{t}, \quad x_8 \leq x_4 - \underline{t}, \quad x_{16} \leq x_{12} - \underline{t} \tag{4.41a}$$

$$0 \leq x_6 \leq x_8, \quad 0 \leq x_7 \leq x_5, \quad x_4 \leq x_2 \leq 0.010, \quad 0 \leq x_3 \leq x_1 \tag{4.41b}$$

where $\underline{t} = 0.0015$ m is the smallest thickness of the specimen. Two other constraints include the implementation of convexity for the large center cut of level 1 and level 2. Figure 4.17 presents an example of deformed configuration after the simulation converges.

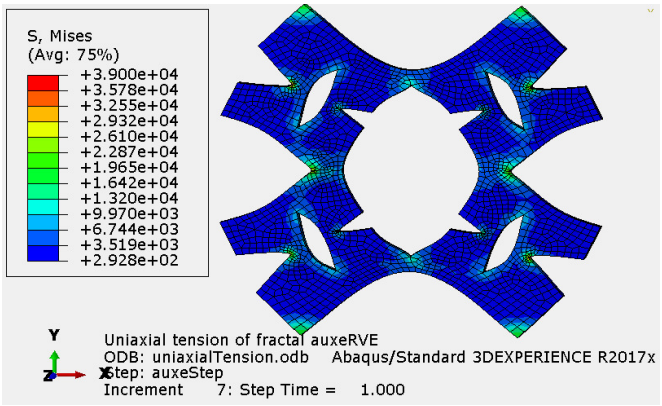


Figure 4.17: An example of uniaxial tension simulation of plane-strain configuration in designing auxetic metamaterials using fractal cut.

Simulation and results

The lower bounds of the continuous variables are (0.25; 3.5; 0.50; 1.75; 8.0; 0.25; 4.0; 0.50; 0.25; 3.5; 0.50; 1.75; 4.0; 0.25; 3.0; 0.50)$\cdot 10^{-3}$. The upper bounds of the continuous variables are (2.00; 6.5; 1.75; 3.00; 9.5; 1.50; 8.0; 1.75; 2.00; 6.5; 1.75; 3.00; 5.5; 1.50; 4.0; 1.75)$\cdot 10^{-3}$.

Two random initial sampling points are created within each cluster. Because the fractal level is fixed at 2, where each fractal level corresponds to one cut motif α or β, 4 clusters are created during the initialization. The initial hyper-parameters θ_i for all i is set at 0.2. The lower and upper bounds for the hyper-parameters θ_i for all i are (0.01, 20).

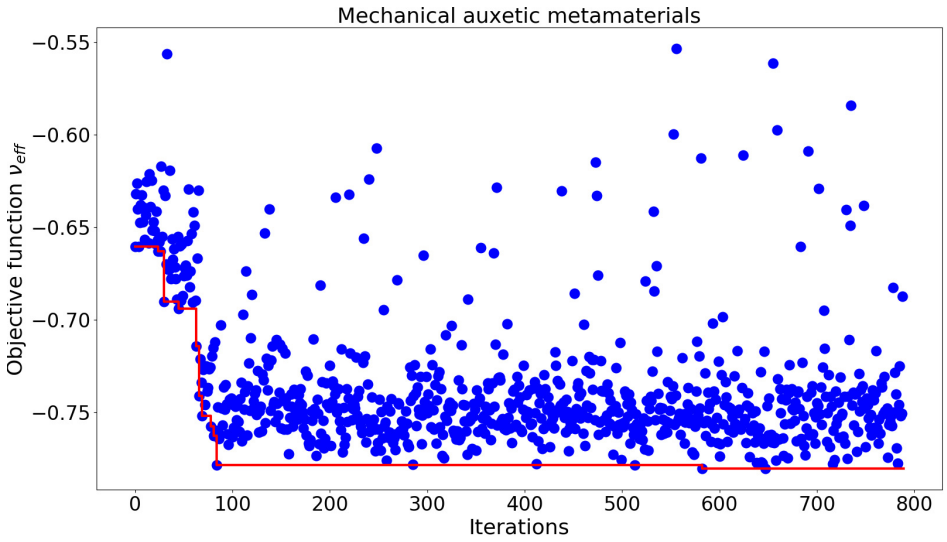


Figure 4.18: Convergence plot of the objective function, which is the effective Poisson's ratio ν_{eff}. Better objective function values are found at the iteration 0, 4, 24, 26, 30, 45, 63, 66, 69, 78, 81, 84, 513, 582, 647, sequentially.

The optimization process is carried out for 790 iterations. Figure 4.18 shows the convergence plot of the optimization process, where the best objective function value ν_{eff} are found in iteration 0, 4, 24, 26, 30, 45, 63, 66, 69, 78, 81, 84, 513, 582, 647, with the value of -0.6603, -0.6605, -0.6628, -0.6628, -0.6902, -0.6941, -0.7143, -0.7410, -0.7517, -0.7576, -0.7627, -0.7784, -0.7785, -0.7802, -0.7804, respectively. The proposed BO shows relatively fast convergence for mid-level dimensionality $d = 16$, thus demonstrating the effectiveness in tackling mix-integer nonlinear optimization problems.

4.6 Discussion

One of the advantages of the proposed BO algorithm is its extension to incorporate discrete variables for nonlinear mixed-integer optimization problems. The discrete variables include both categorical and integer variables, thus can be applied with or without the notion of order. The neighborhood of each cluster is built once during the initialization of the process, and can be customized to adapt to specific user-defined requirements. Additionally, because the neighborhood can be modified and/or defined manually, the independence

between clusters can be achieved by removing the corresponding clusters. Such independence is quite common in the case of categorical variables. However, the optimization performance of the proposed method does not depend on the enumeration of the clusters. We emphasize that if the cluster is ceased to exist, then it can be manually removed, and the cluster indices can be reenumerated manually by a slight modification of Equation 4.12 and Algorithm 1.

The weight computation scheme is devised in such a way that asymptotically, the weight prediction converges to a single GP prediction, by imposing a weight vector which has 0 everywhere, except for a single 1 that corresponds to the corresponding cluster. It is recommended to choose the neighbors carefully. One way to do so is to set a small threshold discrete distance d_{th}, which measures the dissimilarity between clusters based on the discrete tuples, e.g. $d_{\text{th}} \leq 1$, and manually remove clusters that are known to be independent beforehand at the end of initialization. The safest setting is $d_{\text{th}} = 0$, which assumes clusters are completely independent of each other. This setting has some negative effect on the convergence rate, but would eventually reach down to the global optimal solution, and would not be trapped at local optimum.

Here the scalability problem of GP is alleviated, but not completely eliminated. It is noted that the decomposition and weighted average approach has been adopted [157, 158, 159, 160, 117, 1] for continuous variables. The decomposition method for continuous variables is typically referred to as local GP. This approach is promising in tackling the scalability problem. Particularly, in one of our previous studies [1], we have shown that the local GP is one-order computationally cheaper, compared to the classical GP, while maintaining a good approximation error. Nevertheless, further research is required to develop an efficient and robust decomposition scheme for both discrete and continuous variables.

One of the limitations is the scalability with respect to discrete variables. Because of the decomposition scheme, the number of the clusters are the combinatorial possibilities, i.e. the product of the number of choices for each discrete variable, and thus advocating

the sparsity problem in each cluster. To mitigate the undesirable sparsity effect, a Gaussian mixture model that combines all the predictions from neighboring clusters is used to exploit some useful information from the neighborhood. However, significant improvement must be made for the proposed BO algorithm to solve the constrained mixed-integer optimization. As mentioned previously, the mixed-integer optimization problem, in general, is difficult, because it combines the difficulties for both discrete and continuous optimization. Particularly, some discrete optimization problems are NP-complete, such as the traveling salesman problem, knapsack problem, and graph coloring problem, to name a few. An interesting candidate to model the weights as stochastic variables, so that the metaheuristic methodologies can be applied [161].

The clustering and enumeration algorithm described in Algorithm 1 assumes the independence of discrete variables. The Algorithm 1 does not work if the discrete variables are dependent. However, in the case that discrete variables are dependent on each other, manual neighborhood definition of clusters can be introduced manually, and the proposed BO algorithm is functional with the demonstrated efficiency. However, the users must declare the neighborhood of each clusters manually. Strictly speaking, the computational efficiency of the proposed algorithm only depends on the number of clusters, but not the number of discrete variables. If all discrete variables are completely independent of each other, as demonstrated in the two above examples, then the number of clusters is equal to the product of the number of choices for each discrete variable, i.e. $L = \prod p_i$.

Another practical limitation for the proposed BO algorithm for engineering model and simulation is its sequential nature of sampling and search. The main reason is that simulation usually contains much details, and thus demands a considerable amount of computational time. In practice, for high-fidelity and dedicated simulations, one should resort to multi-fidelity or batch-parallel BO for further improvement.

4.7 Conclusion and Future Work

In this paper, we propose a new BO algorithm to solve the nonlinear constrained mixed-integer design optimization problems. In this algorithm, the large dataset is decomposed according to the discrete tuples, in which each discrete tuple corresponds to a unique GP model. The prediction for mean and variance is formulated as a Gaussian mixture model, in which the weights are computed based on the pair-wise Wasserstein distance between clusters. Constraints, which are formulated as a set of inequalities, are included during the optimization process. Theoretical bounds and algorithmic complexity are provided to prove the computational efficiency compared to the classical GP.

The proposed algorithm is demonstrated with two fractal metamaterials design examples, where the mechanical properties are tailored by the hierarchically designed architect. In the first example, the algorithm is used to search for the fractal metamaterial with high-strength and low-density properties, where material selection is considered. In the second example, the algorithm is utilized to design an auxetic metamaterial for flexible and stretchable devices, where the effective Poisson's ratio is chosen as the objective function. For both computational material design examples, constraints are imposed to limit the design space. The proposed algorithm shows a promising performance in solving engineering problems, where high-dimensionality is often an issue.

While several limitations exist, such as scalability for discrete and continuous variables, further research extensions can be made to improve the current methodology, including metaheuristic methodologies for stochastic combinatorial optimization.

CHAPTER 5
OPTIMIZATION UNDER UNCERTAINTY: BATCH-PARALLEL AND CONSTRAINED BAYESIAN OPTIMIZATION

In this chapter, we present a constrained batch-parallel Bayesian optimization (BO) framework, termed pBO-2GP-3B, to accelerate the optimization process for high-dimensional and computationally expensive problems, with known and unknown constraints. Two Gaussian processes (GPs) are simultaneously constructed: one models the objective function, whereas the other models the unknown constraints. The known constraint is penalized directly into the acquisition function. For every iteration, three batches are built in sequential order: the first two are the acquisition hallucination and the exploration batches for the objective GP, respectively, and the third one is the exploration batch for the classification GP. The pBO-2GP-3B optimization framework is demonstrated with three synthetic examples (2D and 6D), as well as a 33D multi-phase solid-liquid computational fluid dynamics (CFD) model for the design optimization of a centrifugal slurry pump impeller.

5.1 Introduction

BO is a surrogate-based black-box optimization technique that models and updates the response surfaces sequentially as the optimization process advances. It has proven to be successful in many different applications including machine learning [162] and design optimization [163] [164]. Several notable advantages of BO, compared to other optimization techniques, include derivative-free, active learning, uncertainty quantification, and mathematically robustness in high-dimensional continuous design space. Yet, the application of classical BO in engineering simulation-based optimization problems is often limited by several factors. First, the computational cost of the traditional BO is expensive, where each functional evaluation is a single simulation, which can take hours to finish. Combined with

the sequential sampling approach in the classical BO, the turnaround time significantly increases. Second, the successful rate of the simulations can be well estimated, but occasionally the simulation will crash due to some unforeseeable factors, e.g. mesh problems, solver issues, or an ill-conditioned matrix. This is a common behavior in developing complex simulations. Therefore, improving these two aspects of simulation-based optimization problems is critical to the optimization problem of complex engineering simulations.

In order to solve optimization problems with complex engineering simulations, in concert with the growth of high-performance computing (HPC) infrastructure, the classical BO has been extended toward different possible directions: batch parallelization, constrained, multi-objective, multi-fidelity, hybrid, and distributed optimization. Batch parallelization for BO is the implementation of selecting a batch, where several concurrent functional evaluations can be performed simultaneously in order to obtain the optimal solution in the shortest wall-clock time. Constrained BO includes constraints in the classical BO formulation. The constraints can be conceptually divided into two main categories: *known* constraints, where the constraints are imposed beforehand and can be evaluated without running the functional evaluator, and *unknown* constraints, where the constraints are unknown *a priori*, and only known once the functional evaluator is invoked. For example, in engineering applications, the known constraints can be thought of as a collection of inequality constraints, where physics-based knowledge is injected to form a set of inequalities upon the design variables. The unknown constraints, on the other hand, can be established once the functional evaluator or the modeling simulation fails to obtain a convergent solution. Progress on improving BO is elaborated in Section 5.2.

With the introduction of the known and unknown constraints, the design space is naturally divided into two regions: feasible and infeasible. The feasible region includes inputs which do not violate any constraints, whereas the infeasible region includes inputs which violate any of these constraints. In order to search for a feasible and optimal input, one needs to predict whether the input is feasible, before attempting to evaluate the functional

value. As a result, the binary classification for feasibility is needed to make such a prediction. Here we propose a novel algorithm pBO-2GP-3B to batch parallelize the classical BO for the general constrained BO problem and demonstrate its applications to complex modeling and simulation methods, such as multi-phase CFD. The known constraint is penalized directly in the acquisition function of the classical BO, whereas the unknown constraints are learned through a classification GP. A novel acquisition function formulation is devised and generalized to incorporate the known and unknown constraints, as well as the trade-off between exploitation and exploration in the classical BO. It is noteworthy that the new acquisition function is applicable for any choice of commonly used acquisition functions, such as probability of improvement, expected improvement, upper-confidence bound. The novel methodology is demonstrated with three synthetic two-dimensional and six-dimensional examples (2D and 6D) and an engineering CFD (33D) example.

5.2 Related work

The BO formulation can be briefly summarized as follows. Given the unknown objective function $y = f(\boldsymbol{x})$, we wish to find the optimal solution that maximizes the objective function $\boldsymbol{x}_{\text{opt}} = arg\,max f(\boldsymbol{x})$. Instead of directly optimizing this unknown black-box function, the GP model is constructed as a response surface. For the classical BO method, the next sample is sequentially chosen to maximize the acquisition function, which can be computed numerically based on the constructed GP response surface. The acquisition function dictates the location of the next sampling design site by reconciling the trade-off between exploration (navigating to the most uncertain region) and exploitation (driving the solution to the best-so-far) in the optimization process. While recently there have been significant advances in devising other acquisition functions, the most popular acquisition functions are probability of improvement (PI) [165], expected improvement (EI) [102] [101] [166], and upper-confidence bound (UCB) [104, 105].

Brochu et al. [167], Shahriari et al. [141], and Jones et al. [168] provide comprehensive

and critical reviews of the classical sequential BO method, including a different acquisition function, as well as its applications. One of the significant drawbacks for the classical BO is the computational bottleneck in optimizing the maximum likelihood function as the number of observations N increases. Estimating hyper-parameters θ of the objective GP involves optimization of the likelihood and computation of the inverse of the covariance matrix, resulting in $\mathcal{O}(N^3)$ algorithmic complexity. In section 5.2.1, several common acquisition functions in BO are reviewed. In section 5.2.2, the relevant work on constrained BO and its incorporation to classical BO is presented. In section 5.2.3, the related work on batch parallel BO is summarized and discussed.

5.2.1 Acquisition function

In the classical BO, a GP model is constructed and updated sequentially as the optimization process advances. Consider a dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, where N is the number of observations, $\boldsymbol{x}_i$ is the input, and y_i is the actual observation in the classical BO settings. Denote $\mu(\boldsymbol{x})$, $\sigma^2(\boldsymbol{x})$, and $\theta(\boldsymbol{x})$ as the posterior mean, the posterior variance, and the hyper-parameters of the GP model, respectively. In practice, the hyper-parameters θ are found by maximizing the log likelihood estimation (MLE) at every iteration over a plausible range between lower and upper bounds of θ. Let $\boldsymbol{x}_{\text{best}} = arg\,max_{\boldsymbol{x}_i} f(\boldsymbol{x}_i)$ be the best sample achieved so far during sequential sampling for a maximization problem, and $\phi(\cdot)$ and $\Phi(\cdot)$ be the probability density function and cumulative distribution function of the standard normal distribution respectively.

The acquisition function for probability of improvement (PI) [165] is defined as

$$a_{\text{PI}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \Phi(\gamma(\boldsymbol{x})), \tag{5.1}$$

where

$$\gamma(\boldsymbol{x}) = \frac{\mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) - f(\boldsymbol{x}_{\text{best}})}{\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta)}, \tag{5.2}$$

indicates the deviation away from the best sample. The acquisition function for expected improvement (EI) scheme [102] [101] [166] [169] is defined as

$$a_{\text{EI}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) \cdot (\gamma(\boldsymbol{x})\Phi(\gamma(\boldsymbol{x})) + \phi(\gamma(\boldsymbol{x})) \tag{5.3}$$

The acquisition function for the upper-confidence bounds (UCB) scheme [104, 105] is defined as

$$a_{\text{UCB}}(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) = \mu(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta) + \kappa\sigma(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^N, \theta), \tag{5.4}$$

where κ is a hyper-parameter describing the acquisition exploitation-exploration balance.

Other forms of acquisition functions include predictive entropy search GP-PES [170] that has been extended to constraint BO [109] and multi-objective BO [171], entropy-search GP-ES [172], and estimation strategy GP-EST [173].

5.2.2 Constrained BO

Constrained BO is a important and natural extension of classical BO. Digabel and Wild [106] summarized and provided a systematic classification taxonomy, QRAK, for derivative-free simulation-based black-box optimization problems. For *known* constraints, the constraints can be realized by either multi-fidelity approach, where the constraint evaluator is considered as a low-fidelity functional evaluator [140], or is penalized directly within the acquisition function [107] inside the sequential sampling search loop for the next query point. Schonlau et al. [174] introduced a generalized EI acquisition function that balances global-local search by incorporating an exponential power with a known constraint penalization scheme. Parr et al. [175] reviewed and compared different constraint handling schemes using synthetic and real-world engineering applications. Picheny et al. [176] proposed an augmented Lagrangian scheme to convert a constrained problem to an unconstrained optimization problem, handling both equality and inequality constraints.

However, the problem is more challenging if the constraints are *unknown*. Several methods have been proposed to incorporate the probability of constraint violation into the acquisition function. Basudhar et al. [177] utilized support vector machines to calculate the feasible probability which is later incorporated in the EI acquisition function. Sacher et al. [178] extended the method of Basudhar et al. [177] by combining classification probability from the least squares support vector machine with the EI acquisition function and proposed another sampling strategy to locate the next sampling point in a sequential manner. Gelbart et al. [108] proposed an entropy search criterion and used the expected improvement acquisition to search for the next sampling location. Hernández-Lobato et al. [109] suggested an alternative mathematical form of acquisition function, to maximize the expected reduction in the differential entropy of the posterior. Gramacy and Lee [179] introduced an integrated expected conditional improvement and increased the number of hyper-parameters to accommodate the class labels. Lee et al. [180] proposed to couple the random forest classifier and maximize the expected constrained improvement, where the new acquisition function is the product of old acquisition function and the predicted probability of the random forest classifier. The main difference between our work and other work, such as Basudhar et al. [177], Sacher et al. [178], and Lee et al. [180], is two-fold. First, our proposed method is batch parallel, whereas other methods are sequential. Second, our proposed acquisition function in Equation 5.7 is generalized and applicable for any probabilistic binary classifier, for example, kNN [181], AdaBoost [182], Random-Forest [183], support vector machine [184] (SVM), least squares support vector machine (LSSVM) [185], and GP.

5.2.3 Batch parallel BO

Typically, to accelerate the optimization process, there are mainly two approaches: accelerating using multiple processor parallel implementation for one simulation, or batch parallelization where each core handles one simulation. One essential observation is that

according to Amdahl's law, the former approach is prone to diminishing returns, where the modeling simulation cannot be accelerated after a certain threshold. Thus the later approach, batch parallel BO is more cost-effective and appealing toward computationally expensive engineering models and simulations.

The core idea of batch parallel BO is simply based on the concept of active learning, in which the optimization history is used to reconstruct the response surface and approximate the objective function statistically. Therefore, it is possible to perform multiple functional evaluations at the same time for a single batch, and update the history once the batch is finished. To search for the new points with unknown objective functions, one can rely on the Gaussian probability prediction of the outputs and condition the acquisition function on this Gaussian distribution. For example, Snoek et al. [169] proposed an integrated acquisition function based on Monte Carlo sampling, called GP-EI-MCMC, to construct a parallel batch, which is conditioned on the Gaussian probability of the observation. Ginsbourger et al. [186] [187] [188], Roustant et al. [189] suggested a multi-points expected improvement q-EI, in which the EI function is conditioned on the Gaussian observation. Marmin et al. [190] [191] provided analytical formula for gradient-based q-EI extension. Letham et al. [192] extended the q-EI framework toward noisy and constrained problems and performed the experiments at Facebook. Wang et al. [193] and Wu and Frazier [194] (q-KG) both suggested a more numerically efficient approach, based on infinitesimal perturbation analysis, to compute the acquisition function of the gradient-based q-EI extension. Shah and Ghahramani [195] extended the predictive entropy search GP-PES [170] toward parallel batch selection policy GP-PPES. Azimi et al. [196] [197] proposed a simulation matching scheme GP-SM [196] and coordinated matching scheme GP-BCM [197] to select input batches that closely match their expected behavior (GP-SM) or the sequential distribution after a certain amount of step (GP-BCM). Azimi et al. [198] also proposed an alternative switching between sequential and parallel mode for BO and analyzed the theoretical bound for the hybrid batch BO method. Desautels et al. [199] introduced a hallucination

scheme for batch selection policy GP-BCUB and its adaptive variant GP-AUCB, where the main acquisition is GP-UCB [104] [105], and showed the theoretical bound of the proposed method. The hallucination scheme for a single GP is based on the assumptions of the posterior mean as the prediction during the current batch parallel iteration, and the posterior variance as zero at the particular input location of the batch. The hallucination scheme is almost identical with the kriging believer heuristic in Ginsbourger et al. [187], except that in the hallucination scheme, the posterior variance $\sigma^2(\boldsymbol{x})$ is also updated following the hallucination. Contal et al. [200] proposed a modified version of GP-BUCB, termed GP-UCB-PE, where only one location is selected via GP-UCB acquisition function, and the rest of the batch is configured to locations where the updated posterior variance is maximized. Both GP-BUCB and GP-UCB-PE are constructed based on a crucial observation that the posterior variance only depends on the input locations, but not the actual output. The theoretical bound of GP-UCB-PE is proven to be better than that of GP-BUCB by an factor of $\sqrt{B_{\text{batch}}}$, where B_{batch} is the size of the batch. González et al. [201] proposed GP-BBO-LP method, which assumes the objective function is Lipschitz continuous, infers the Lipschitz constant directly from the GP, and modifies the acquisition function with a local penalizer and a differentiable transformation functions. Kathuria et al. [202] and Wang et al. [203] offered a batch selection policy via determinantal point process GP-DPP, and proved the expected regret bound of DPP-SAMPLE is less than the regret bound of GP-UCB-PE [200]. Kathuria et al. [202] showed that GP-UCB-PE is a special case as DPP-MAX, where the maximization rule is done via a greedy selection rule, and suggested that GP-PPES [195] performs better than GP-BUCB [199] and GP-UCB-PE [200] approach, and that GP-UCB-PE [200] performs better than GP-SM approach [196]. There are also concerns that GP-UCB-PE and GP-BUCB are too greedy in the batch selection process, and thus prone to be non-optimal with respect to the "immediate overconfidence" measure [202]. Rontsis et al. [204] proposed an alternative acquisition function GP-OEI, where the lower and upper bounds are computationally tractable in high-dimensional space

and showed its numerical robustness over q-EI. Nguyen et al. [205] proposed budgeted batch BO, termed GP-B3O, which utilizes the infinite Gaussian mixture model to automatically identify the number of peaks in the underlying acquisition function, and adapts the batch size accordingly based on the approximated acquisition function. Daxberger and Low proposed a novel distributed batch GP-UCB, dubbed DP-GP-UCB [206], to jointly optimize a batch of inputs, as opposed to selecting the inputs of a batch one at a time, and still preserve the scalability in the batch size.

5.3 Methodology

In this chapter, we propose a two-GP batch-parallelization approach to solve the problem, in which one GP models the objective functions, which is referred to as the objective GP, whereas the other models the binary output of the classification, which is referred to as the classification GP. For the objective GP model, the infeasible data points are also included by the interpolation that occurs at each iteration, where interpolation process is performed in two steps. In the first step, *only* the feasible data points are used to construct the objective GP model. In the second step, the constructed objective GP is used to predict the output at infeasible data points. Then the GP model uses the predicted posterior mean $\mu_{\text{objective}}(\boldsymbol{x})$ at those infeasible data points to update again to reflect the true posterior variance in the objective GP, i.e. $\sigma^2_{\text{objective}}(\boldsymbol{x}) = 0$ at the infeasible locations $\boldsymbol{x}$. This interpolation process occurs for every iteration.

Parallel in BO is performed through batch parallelization method, in which a batch of size B_{batch} in every iteration is decomposed into three batches: the first batch for the acquisition hallucination batch of size $B_{\text{acquisition}}$ for the objective GP, the second batch for the exploration of size B_{explore} for the objective GP, and the third batch for the binary classification size B_{classif} of the classification GP, *sequentially* with respect to the batch order. All three batches are constructed by hallucinating the GP models at each iteration, sequentially with respect to each sampling point in the batch, where hallucination means that the obser-

vation is assumed to be exactly the posterior mean of the objective GP $\mu_{\text{objective}}(\boldsymbol{x})$ for one iteration, but later on will be corrected once the batch is finished. Also, the feasible probability of the input location $\boldsymbol{x}$ is assumed to be 1, and the posterior variances for both GPs are updated on the chosen sampling location $\boldsymbol{x}$, i.e. $\sigma^2_{\text{objective}}(\boldsymbol{x}) = \sigma^2_{\text{classification}}(\boldsymbol{x}) = 0$. The input locations in the exploration batch are selected where the updated posterior variance σ^2 is maximal for both GPs. The infeasible data points are assigned as an interpolated value and subjected to change after each iteration. To incorporate the feasibility classifier, the acquisition is reformulated to condition on the probabilistic prediction, and thus become the expected acquisition function.

The technical contribution of this chapter is three-fold. First, the advantage of batch BO is extended to incorporate the constrained optimization problems, based on the premise of a HPC infrastructure, using an expected acquisition function that is conditioned on the predicted feasibility from the probabilistic feasibility classifier. Second, the feasibility classifier is continuously improved by the pure exploration batch, so that the feasibility classifier is forced to learn in the unknown regions to improve its predictability performance. Third, a new acquisition is proposed to combine two GPs models.

5.3.1 Constraints and Feasibility Classification

In this chapter, we are concerned with both types of constraints: *known* inequality constraints, which are known in advance and usually physics-based modeled, and *unknown* constraints, which are only known when the functional evaluation actually ocurrs. The infeasibility function of a design can be represented as a set of inequalities $\lambda(\boldsymbol{x}) \leq \boldsymbol{c}$, which in turn can be interpreted as a constrained optimization problem. To penalize the infeasible design, the feasibility checking function is embedded within the acquisition function, to assign zero improvement to all infeasible points if the sampling location $\boldsymbol{x}$ does not satisfy the *known* constraints $\lambda(\boldsymbol{x}) \leq \boldsymbol{c}$. Mathematically, $a(\boldsymbol{x}) = 0$ if there exists an index i such that one of the constraints is unsatisfied $\lambda_i(\boldsymbol{x}) > c_i$. In classical BO at each optimization

iteration, the next sampling point is chosen to maximize the acquisition

$$\boldsymbol{x}_{\text{next}} = arg\,max_{\lambda(x)\leq \boldsymbol{c}} a(\boldsymbol{x}; \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}, \theta), \tag{5.5}$$

where N is the number of observations, and θ is the hyper-parameters of the metamodel.

We propose a variant acquisition function conditioned on the probabilistic prediction of binary GP classification. The acquisition function is updated according to the probabilistic classification augmented in the classification model to obtain the EI function

$$\mathbb{E}[a(\boldsymbol{x})] = 0 \cdot Pr(\text{clf}(\boldsymbol{x}) = 0) + a(\boldsymbol{x}) \cdot Pr(\text{clf}(\boldsymbol{x}) = 1) = a(\boldsymbol{x}) \cdot Pr(\text{clf}(\boldsymbol{x}) = 1), \tag{5.6}$$

where clf$(\cdot)$ denotes the binary probabilistic classifier, and $Pr(\cdot)$ denotes the probability mass function (pmf) of the design variables, $Pr(\text{clf}(\boldsymbol{x}) = 0)$ is the probability that the design variable $\boldsymbol{x}$ is infeasible, and $Pr(\text{clf}(\boldsymbol{x}) = 1)$ is the probability that the design variable $\boldsymbol{x}$ is feasible.

5.3.2 Batch parallelization

A part of this high-throughput BO framework is built on the premise of a HPC. The idea is to bring down the computational runtime of the optimization of expensive high-fidelity simulations through increased parallelism in HPC. For better computational resource allocation in HPC usage, in the extreme case, the diminishing return in the multi-core simulations can be avoided by parallelizing all the simulation in such a serial/sequential manner. That is, m processors can be used to perform m different simulations, where each processor corresponds to a simulation with different input parameters, yielding no diminishing return in Amdahl's law [207]. The calculations of posterior variances σ^2's do not depend on the observations for both GPs, therefore the exploration sampling points are determined after the acquisition hallucination. Desautels et al. [199] hallucination scheme is adopted during the batch selection process.

The interpolation process can be considered as hallucinating the objective GP at infeasible locations, by assigning the posterior means observations, after constructing the objective GP using data only at feasible locations. The purpose of introducing the posterior means as observations per iteration is to truly reflect the posterior variance $\sigma^2_{\text{objective}}(\boldsymbol{x})$ of the objective GP, particularly at infeasible locations. That is, the objective GP is aware of the locations where observations have been failed to be obtained. In order to do so, the objective GP assumes the posterior mean $\mu_{\text{objective}}(\boldsymbol{x})$ as the observation per iteration at infeasible location, for the same hyper-parameters θ of the objective GP. As the optimization advances in a batch sequential manner, the hyper-parameters θ of the objective GP changes accordingly, and the interpolation process is repeated again. The mechanism does not interfere in locating the next sampling point, because Equation 5.7 is used to do so, where the feasible probability is predicted using an alternative binary classifier. Otherwise, the posterior variance $\sigma^2_{\text{objective}}(\boldsymbol{x})$ surface is considerably large at infeasible locations, thus not truly reflected at those locations. The incorrect $\sigma^2_{\text{objective}}(\boldsymbol{x})$ would have an impact on the second term of Equation 5.7, misleading the algorithm into infeasible regions, which we seek to avoid.

Acquisition hallucination batch for objective GP

We describe the point selection process for the first acquisition hallucination batch in three batches, denoted as $B_{\text{acquisition}}$. The "hallucination" term is chosen to explain the effect of assuming an observation in the current batch, which is actually unknown, but is assumed to be known, right at the moment of the batch construction, and this observation will be further corrected at the end of the current batch iteration before moving to the next batch iteration. Technically, all PI, EI, and UCB acquisition functions feature a trade-off between exploitation and exploration within their analytical formulations. In this process, the point is selected by maximizing the acquisition function, which could be PI, EI, or UCB function. Then, the objective GP is hallucinated to temporarily register the selected point

$\boldsymbol{x}$ and its posterior mean $\mu_{\text{objective}}(\boldsymbol{x})$ as the actual observation until the end of this iteration. The posterior variance of the objective GP is updated to reflect the posterior variance $\sigma^2_{\text{objective}}(\boldsymbol{x}) = 0$ at the selected point $\boldsymbol{x}$. Also, the classification GP is hallucinated to realize the sampling point as feasible until the end of this iteration. This hallucination process happens sequentially within the first acquisition hallucination batch, and stops when the number of selected points reaches $B_{\text{acquisition}}$.

Pure exploration batch for objective GP

After the acquisition hallucination batch is constructed, we shift our focus to the second batch, which is the pure exploration batch for the objective GP. The second batch is referred to as B_{explore}. Using the same hallucination approach, the objective GP and the classification GP register the sampling point and the posterior mean μ as the actual observation until the end of the iteration. The points are selected where the posterior variance σ^2 of the objective GP are maximized. After each point in the second batch is selected, both of the hallucinated GPs are updated to reflect the new posterior variances $\sigma^2_{\text{objective}}$ and $\sigma^2_{\text{classification}}$. The process repeats until the number of selected points in the second batch reaches B_{explore}.

Pure exploration batch for classification GP

The last batch focuses on the convergence of the classification GP and is denoted as $B_{\text{exploreClassif}}$. Classification in high-dimensional space generally suffers from the curse of dimensionality, and thus, this batch is devised to force the classification GP to learn in the most uncertain region. In this batch, the sampling points are chosen where the posterior variance $\sigma^2_{\text{classification}}$ of the classification GP are maximized. After each point in the third batch is selected, the classification GP is updated to reflect the new posterior variance $\sigma^2_{\text{classification}}$. The process repeats until the number of selected points in the third batch reaches $B_{\text{exploreClassif}}$.

5.3.3 Acquisition function

The final form of the acquisition function is modified as

$$a^*(\boldsymbol{x}) = \mathbb{E}[a(\boldsymbol{x})] = I(\lambda(\boldsymbol{x}) \leq \boldsymbol{c}) \cdot a(\boldsymbol{x}) \cdot Pr(\text{clf}(\boldsymbol{x}) = 1), \tag{5.7}$$

conditioned on the classifier prediction probability and inequality constraint $\lambda(\boldsymbol{x}) \leq \boldsymbol{c}$, where $I(\cdot)$ is the *known* constraint indicator function, which is embedded within the acquisition function, $a(\boldsymbol{x})$ is the typical acquisition function, which can be PI, EI, UCB, or other types. $Pr(\text{clf}(\boldsymbol{x}) = 1)$ is the predicted probability in which the input $\boldsymbol{x}$ is feasible. The novel acquisition function is composed as the product of the classical acquisition function and two feasibility constraints: one is known, and the other is unknown. It is noteworthy that Equation 5.7 is applicable to any probabilistic binary classifier, i.e. the classifier is not restricted to be GP. The proposed algorithm pBO-2GP-3B seeks next sampling point by maximizing this acquisition function.

5.3.4 Algorithm outline

Algorithm 3 presents the overview of the pBO-2GP-3B formulation, in which a batch is selected based on two criteria: acquisition hallucination batch and pure exploration batch. The later criteria applies for two GPs, respectively, thus form a three-batch approach.

5.4 Synthetic examples

Technically, handling the known constraints is much easier than handling the unknown constraints. In the pBO-2GP-3B formulation, both of them would yield zero value in the acquisition function, but the penalization scheme for the known constraints would make the optimization of the acquisition function more complex. Therefore, only unknown constraints are used in these three synthetic examples. Covariance matrix adaptation evolution strategy (CMA-ES) [144] is used as an auxiliary optimizer to find the location where the

Algorithm 3 pBO-2GP-3B algorithm.

Input: dataset $\mathcal{D}_n$ consisting of input, observation, feasibility $(\boldsymbol{x}, y_i, c_i)_{i=1}^n$
Input: objective GP $(\boldsymbol{x}, y_i)_{i=1}^n$, and classification GP $(\boldsymbol{x}, c_i)_{i=1}^n$

for $n = 1, 2, \ldots,$ **do**
  construct the objective GP
    collect feasible data subset $(\boldsymbol{x}_i, y_i, c_i = \text{feasible})\}_{i=1}^N$
    construct the objective GP, $\mathcal{GP}_{\text{objective}}(\boldsymbol{x}_i, y_i | c_i = \text{feasible})$, for feasible points
    hallucinate the objective GP, i.e. $y_i \leftarrow \mu_i$, at infeasible points $c_i = \text{infeasible}$
    reconstruct the objective GP using both feasible and infeasible points
  construct the classification GP, $\mathcal{GP}_{\text{classif}}(\boldsymbol{x}_i, c_i)$
  select a batch $\mathcal{B}$ of size $B_{\text{batch}} = B_{\text{acquisition}} + B_{\text{exploration}} + B_{\text{classif}}$ points
    acquisition hallucination: hallucinate 2GPs and select $B_{\text{acquisition}}$ points
    exploration: hallucinate 2GPs and select B_{explore} points where $\sigma^2_{\text{objective}}$ is maximized
    classification: hallucinate $\mathcal{GP}_{\text{classif}}$ and select B_{classif} points where $\sigma^2_{\text{classification}}$ is maximized
  query objective function for objective $y_{n+1}^{(B)}$ and feasibility $c_{n+1}^{(B)}$ for the current batch
  augment dataset $\mathcal{D}_{n+1} = \{\mathcal{D}_n, (\boldsymbol{x}_{n+1}^{(B)}, y_{n+1}^{(B)}, c_{n+1}^{(B)})\}$
end for

acquisition function is maximized.

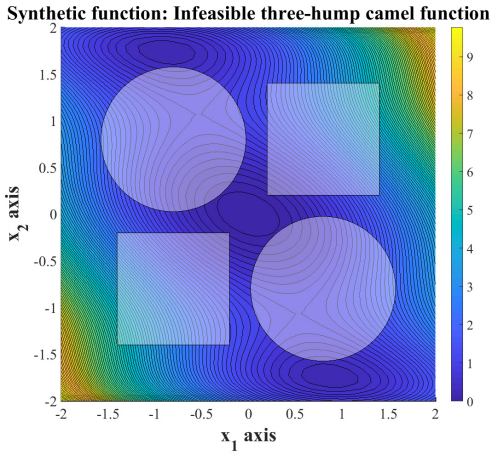
5.4.1 2D Three-hump camel function

In this example, the proposed algorithm is tested on a 2D three-hump camel function on the domain $[-2, 2] \times [-2, 2]$, where the function is

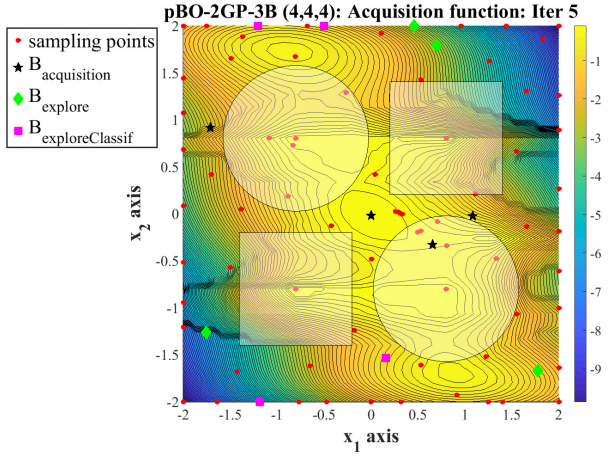
$$f(\boldsymbol{x}) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2. \tag{5.8}$$

The global minimum of this function is $f(\boldsymbol{x}^*) = 0$ at $\boldsymbol{x}^* = (0, 0)$. There are four disjoint infeasible regions, including two circles and two squares. The radius of the circle is $\sqrt{0.60} = 0.7746$ for both circles. The dimension of the squares are 1.20 for both squares. There is no known constraint in this numerical problem. Figure 5.1a shows the contour plot of the unknown constrained three-hump camel function, where the infeasible regions, including two circles and two squares, are shaded. Figure 5.1b presents a batch selection during iteration 5. Readers are referred to color version online. Four sampling locations of

the first batch, $B_{\text{acquisition}}$, are plotted as (black) stars. Four sampling locations of the second batch, B_{explore}, are plotted as (green) diamonds. Four sampling locations of the third batch, $B_{\text{exploreClassif}}$, are plotted as (magenta) squares in Figure 5.1b. As expected, the optimization convergence generally advances due to the sampling locations of the first batch $B_{\text{acquisition}}$. In this case, the second sampling location of the batch $B_{\text{acquisition}}$ is very close with the global optimum of the function.



(a) Contour plot and filled infeasible regions of the 2D unimodal three-hump camel function domain. The infeasible regions include two circles and two squares, and are filled to distinguish with the feasible regions.

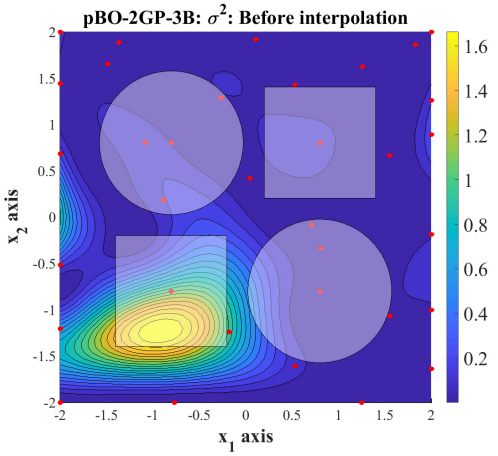


(b) Sampling locations during a batch selection process at iteration 5 for the unknown constrained three-hump camel function, showing 4 sampling locations in each batch.

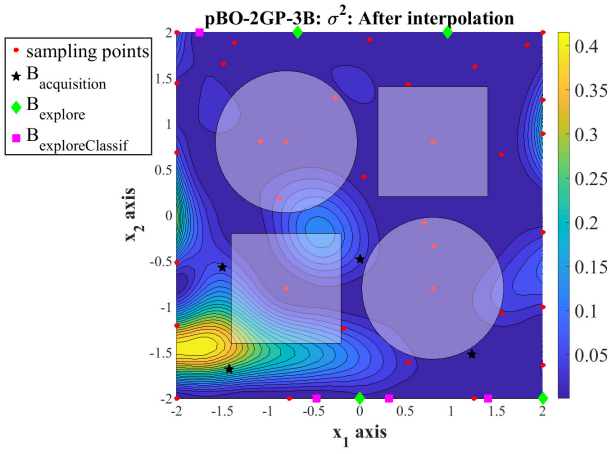
Figure 5.1: Three-hump camel function and its infeasible-feasible domain to test pBO-2GP-3B algorithm.

Figure 5.2a and Figure 5.2b show the posterior variance of the objective GP $\sigma^2_{\text{objective}}$, before and after the interpolation process, respectively. The $\sigma^2_{\text{objective}}$ is significantly lower after the interpolation process, by hallucinating the objective GP at infeasible locations. Without the interpolation process, the posterior variance $\sigma^2_{\text{objective}}$ is large at infeasible regions, promoting the acquisition function to explore more at these regions. As a result, the algorithm would be misguided into infeasible regions.

Figure 5.3 shows the convergence of the binary classification problem using the GP classifier. Figure 5.3a shows the probability of feasibility with 10 data points, whereas Fig-



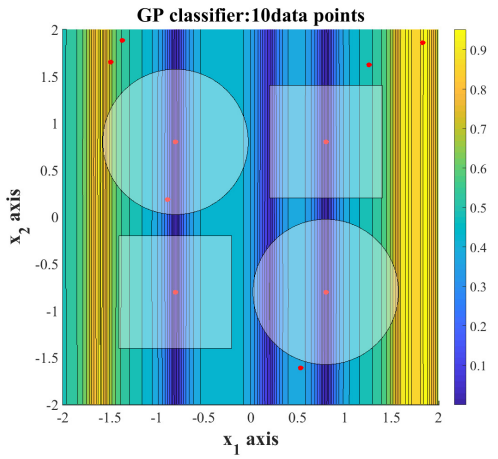
(a) Posterior variance of the objective GP, $\sigma^2_{\text{objective}}$, before interpolation process for infeasible sampling locations.



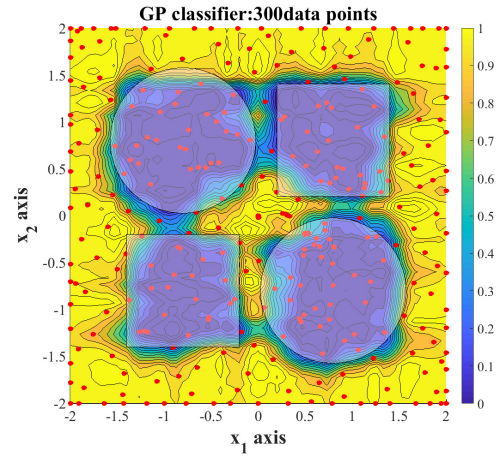
(b) Posterior variance of the objective GP, $\sigma^2_{\text{objective}}$, after interpolation process for infeasible sampling locations.

Figure 5.2: Illustration of the interpolation process, which hallucinates the objective GP at infeasible sampling locations, to truly reflect the posterior variance $\sigma^2_{\text{objective}}$ at the infeasible locations. Without the interpolation process, the posterior variance $\sigma^2_{\text{objective}}$ is large at infeasible regions, thus promoting sampling in infeasible regions, which is undesirable.

ure 5.3b shows the probability of feasibility with 300 data points. In order to successfully solve the unknown constrained optimization problem using the binary classifier, the binary classifier must converge to a certain extent, so that the sampling locations can converge to the global optimum.



(a) GP classification with 10 data points.



(b) GP classification with 300 data points.

Figure 5.3: Convergence of GP classifier for binary classification problem: feasible or infeasible.

For the three-hump camel function, the $B_{\text{acquisition}}$, B_{explore}, $B_{\text{exploreClassif}}$ are set to 4, 4, and 4, respectively. Figure 5.4 shows the convergence plot of pBO-2GP-3B method, where 12 iterations with batch parallelization are performed, in addition to 6 initial sampling points. The connecting solid line denotes the best solution so far as the optimization advances. The UCB acquisition function is used as the main acquisition $a(\boldsymbol{x})$ in the Equation 5.7. For the objective GP, the initial hyper-parameters θ are set as $(1, 1)$, whereas the lower and upper bounds for θ are $(10^{-2}, 10^{-2})$ and $(1, 1)$, respectively. The Gaussian kernel is used to construct the objective GP. For the classification GP, the initial hyper-parameters θ is set as $(1, 1)$, whereas the lower and upper bounds for θ are $(10^{-2}, 10^{-2})$ and $(2 \cdot 10^{1}, 2 \cdot 10^{1})$, respectively. The hyper-parameters θ are obtained by the maximum likelihood estimation method. The remaining learning parameters of the toolbox were left to the default value. The exponential kernel is used to construct the classification GP. In iteration 73, pBO-2GP-3B converges to the solution of $f(0.00208764, 0.00180051) = 0.00001572$, showing the fast convergence rate where both the GPs approximate the unknown function very well.

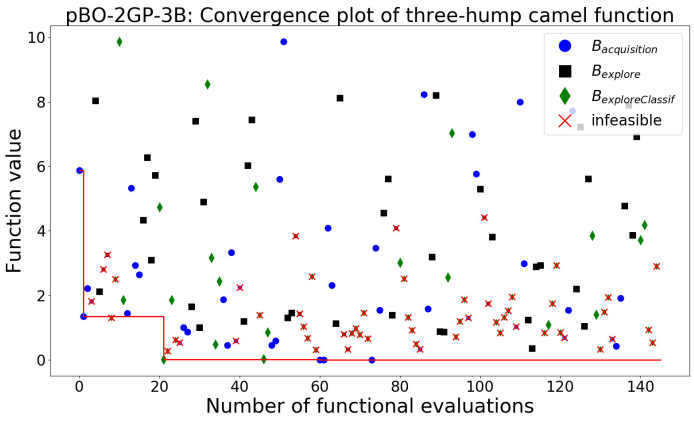


Figure 5.4: Convergence plot of pBO-2GP-3B on the 2D unimodal three-hump camel function. The feasible data points are denoted as solid circles, whereas infeasible data points are denoted as red crosses. The functional value of infeasible data points are not available, but are evaluated in this plot to aid the visualization.

Figure 5.5 compares the proposed pBO-2GP-3B method with other sequential BO algorithms, including GP-EI-LSSVM by Sacher et al. [178], GP-EI-SVM by Basudhar et

al. [177], and GP-EI-RandomForest by Lee et al [180]. Many other binary classifiers, including kNN [181], AdaBoost [182], RandomForest [183], support vector machine [184] (SVM), and least squares support vector machine (LSSVM) [185], are implemented to compare the numerical performance to the proposed pBO-2GP-BO method. Other parallel, unknown constrained BO algorithms are not readily available. Thus, we limit the scope of comparison to other sequential BO methods, which are easy to implement. It is shown that compared to other sequential BO methods, the proposed parallel pBO-2GP-3B converges faster in terms of the computational runtime because of the increasing parallelism, which is controlled by the size of the batches. It is also shown that the UCB acquisition function performs better, compared to EI acquisition function in the three-hump camel function.

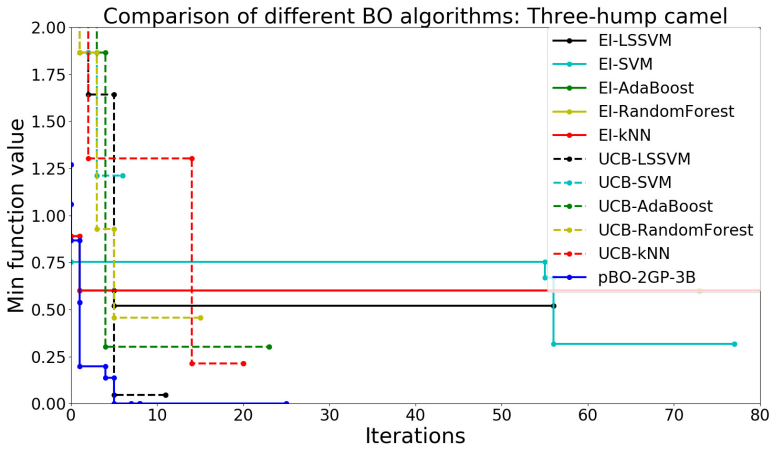


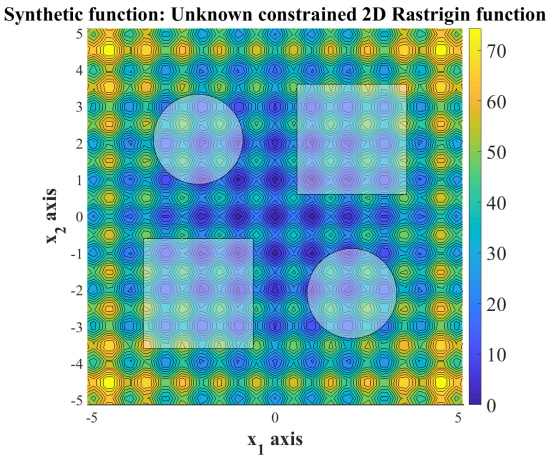
Figure 5.5: Comparison of different BO algorithms using two acquisition functions and various binary classifiers: three-hump camel function.

5.4.2 2D Rastrigin function

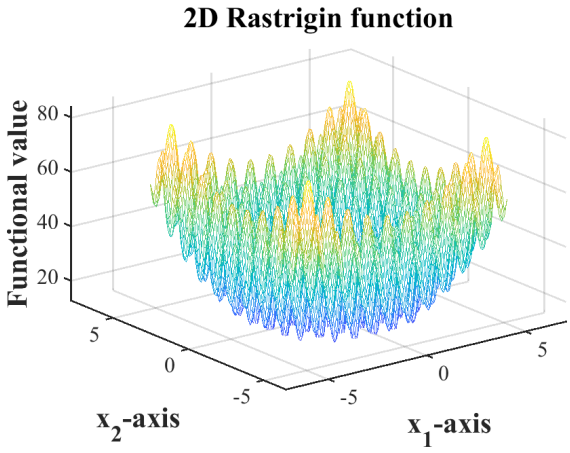
In this example, the proposed algorithm pBO-2GP-3B is tested on a 2D highly multi-modal Rastrigin function [208] [209] on the domain $[-5.12, 5.12] \times [-5.12, 5.12]$, where the objective function is defined as

$$f(\boldsymbol{x}) = 20 + \sum_{i=1}^{2} [x_1^2 - 10\cos(2\pi x_i)]. \tag{5.9}$$

The global minimum of the objective function is $f((0,0)) = 0$. The feasible and infeasible regions are defined similarly to the previous testing example. The dimensions of the squares are 1.50, whereas the diameters of the circles are $\sqrt{1.50} = 1.225$. Figure 5.6a presents the contour plot with infeasible regions shaded, whereas Figure 5.6b shows the 3D visualization of the highly multi-modal 2D Rastrigin function.



(a) Feasible and infeasible regions of the 2D Rastrigin function domain. The infeasible regions include two circles and two squares, and are filled to distinguish with the feasible regions.



(b) 3D visualization of 2D Rastrigin function.

Figure 5.6: 2D Rastrigin function and its infeasible-feasible domain to test pBO-2GP-3B algorithm.

The $B_{\text{acquisition}}$, B_{explore}, $B_{\text{exploreClassif}}$ are set to 5, 1, and 1, respectively. Figure 5.7 shows the convergence plot, where 12 iterations with batch parallelization are performed, in addition to 6 intial sampling points. The connecting solid line denotes the best solution so far as the optimization advances. The EI acquisition function is used as the acquisition function $a(\boldsymbol{x})$ in the Equation 5.7. For the objective GP, the initial hyper-parameters θ is set as $(1,1)$, whereas the lower and upper bounds for θ are $(10^{-2}, 10^{-2})$ and $(1,1)$, respectively. The Gaussian kernel, mathematically expressed as [143] $k_{\text{Gaussian}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\theta_j(\boldsymbol{x}_j - \boldsymbol{x}'_j)^2\right)$, is used to construct the objective GP. For the classification GP, the initial hyper-parameters θ is set as $(1,1)$, whereas the lower and upper bounds for θ are

$(10^{-2}, 10^{-2})$ and $(2 \cdot 10^{1}, 2 \cdot 10^{1})$, respectively. The exponential kernel, mathematically expressed as [143] $k_{\text{exp}}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\theta_j |\boldsymbol{x}_j - \boldsymbol{x}'_j|\right)$, is used to construct the classification GP. At the beginning phase of the optimization process, pBO-2GP-3B encounters a substantial number of functional evaluations in infeasible regions, and gradually converges after that. At the iteration 53, pBO-2GP-3B converges to $f(-0.97776279, 0.98550945) = 2.06611835$, showing a good convergence rate for complex functions, as in the case of 2D Rastrigin function.

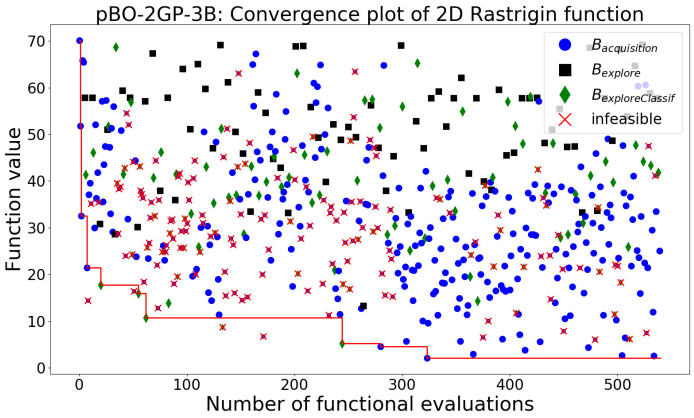


Figure 5.7: Convergence plot of pBO-2GP-3B on the 2D highly multi-modal Rastrigin function. The feasible data points are denoted as solid circles, whereas infeasible data points are denoted as red crosses. The functional value of infeasible data points are not available, but are evaluated in this plot to aid the visualization.

Figure 5.8 compares the proposed parallel pBO-2GP-3B, again with other BO methods. The used binary classifiers are kNN, AdaBoost, RandomForest, SVM, and LSSVM. Both acquisition functions EI and UCB are utilized. The comparison shows that because of the parallelism, the proposed pBO-2GP-3B method performs relatively well compared to other sequential BO methods.

5.4.3 6D Rastrigin

In this example, the proposed algorithm is tested on 6D Rastrigin function on the domain $x_i \in [-5.12, 5.12]$ for all $i = 1, 2, \cdots, 6$. The global minimum of this function

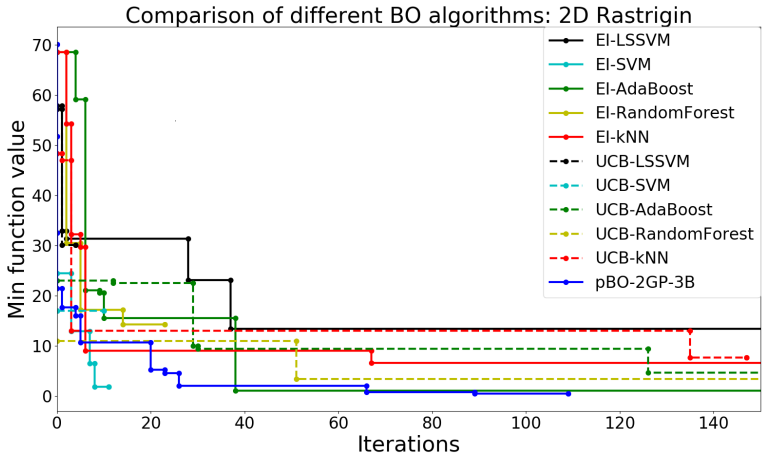


Figure 5.8: Comparison of different BO algorithms using two acquisition functions and various binary classifiers: 2D Rastrigin function.

is $f(x^*) = 0$, where $x^* = (0, 0, 0, 0, 0, 0)$. No known constraint is imposed, whereas six unknown constraints with l_2-norm are embedded blindly to the functional evaluators. Mathematically, they can be described as

$$g_i(\boldsymbol{x}) = \|\boldsymbol{x} - 2.56\boldsymbol{v}_i\|_2 \geq 5, \quad i = 1, \cdots, 6, \tag{5.10}$$

where $\boldsymbol{v}_i = [-1, \cdots, 1 \cdots, -1]$ is a vector where i-index element is 1, and other elements are -1.

Figure 5.9 shows the convergence plot of pBO-2GP-3B for the 6D Rastrigin function. The connecting solid line denotes the best solution so far as the optimization advances. EI acquisition function is used to find the next sampling points in the hallucination batch. The $B_{\text{acquisition}}$, B_{explore}, $B_{\text{exploreClassif}}$ parameters in this example are set to 6, 6, and 6, respectively. In iteration 18 (functional evaluation 308), pBO-2GP-3B converges to $f(-0.97884957, 1.97082269, 1.03444007, 1.88768059, 0.19720165, 2.07607401) = 24.56607326$, again, showing a good convergence rate for intermediate dimensionality and a highly complex function.

Figure 5.10 presents the comparison plot of different BO algorithms for the 6D Rastrigin function, showing that the pBO-2GP-3B algorithm performs on a par with other

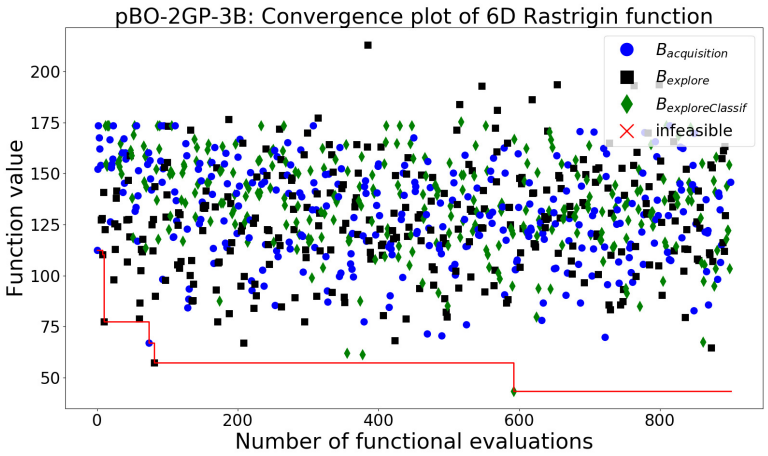


Figure 5.9: Convergence plot of pBO-2GP-3B on 6D Rastrigin function. The feasible data points are denoted as solid circles, whereas infeasible data points are denoted as red crosses.

sequential BO algorithms. The same classification methods and acquisition functions are used in this comparison.

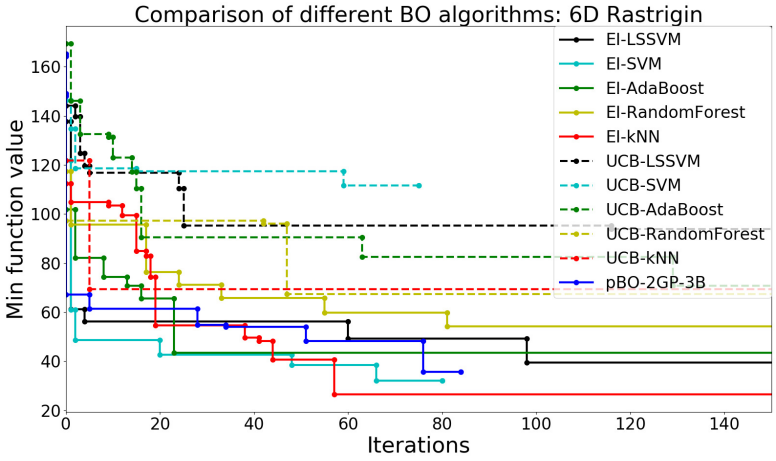


Figure 5.10: Comparison of different BO algorithms using two acquisition functions and various binary classifiers: 6D Rastrigin function.

5.5 Design optimization of slurry pump impeller

In this section, a case study of CFD simulation to assess the erosion wear rate of the impeller in the slurry pump [210] is used to demonstrate the functionality of the proposed pBO-2GP-3B method in design optimization. Here, a multi-phase solid-liquid CFD model is treated as a black-box function mapping from the design space to the predicted wear

performance at certain operating conditions.

In the design space, the geometry of the impeller is modeled using two Bézier patches, where the 3D cylindrical $z-$, $r-$, and $\theta-$coordinates of the Bézier control points are the inputs of the black-box function. For each functional evaluation, a novel geometric model of the impeller is constructed using Bézier patch formulation. Then the CFD erosion wear model is simulated as the functional evaluator. Finally, in the post-process, an average wear quantity is computed on the impeller vanes (suction and pressure sides) to assess the average wear rate.

The CFD erosion wear model for slurry pump impeller is extended and further developed from previous work [211] to capture the multi-size particulate flow. An Eulerian-Eulerian mixture model is used, where volume and time averaged governing equations are derived for the continuity and momentum of the mixture and the individual species in the particle size distribution.

To that end, the goal of this engineering example is to search for the optimal design of slurry pump impeller, which minimizes the predicted erosion wear rate, which is obtained by the post-process of the predictive CFD erosion wear simulation.

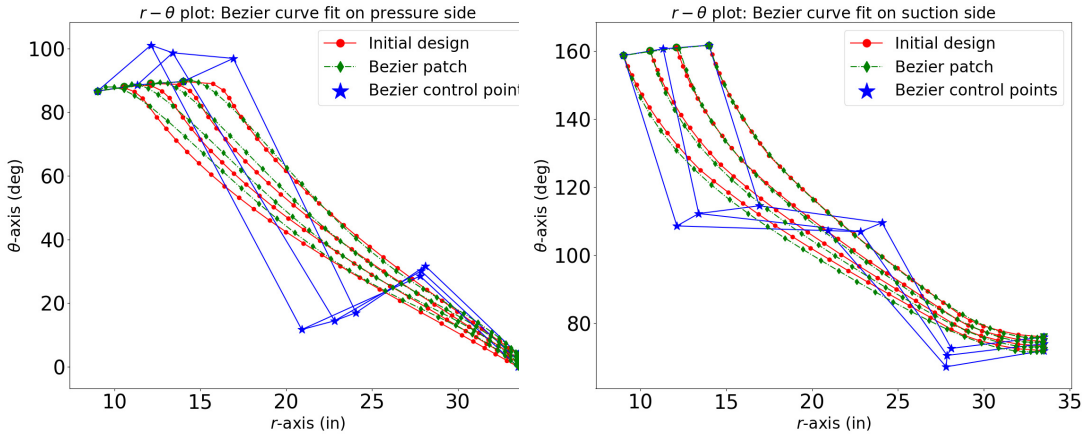
5.5.1 Parameterization of design variables

The geometry of the impeller vane is discretized and modeled using two Bézier patches, one for the pressure face, and the other for the suction face,

$$\vec{p}(u,v) = \sum_{i=0}^{m}\sum_{j=0}^{n} B_i^m(u) B_j^n(v) \vec{b}_{i,j}, \tag{5.11}$$

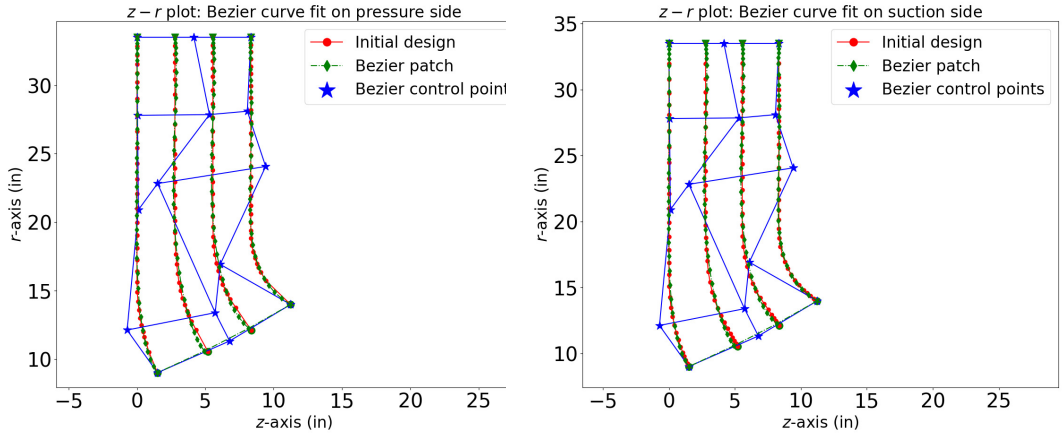
where $B_i^m(u)$ is the Bernstein polynomial of degree m evaluated at $0 \leq u \leq 1$, and $\vec{b}_{i,j}$ is the (i,j)-th control point' coordinates of the Bézier patch. 3D cylindrical coordinate system is used instead of Cartesian coordinate system due to the nature of angular rotation in impeller. Along the vane direction, m Bézier control points are used, whereas in the

transverse direction, n Bézier control points are used. This approach results in $2(m+1)(n+1)$ control points for both pressure and suction faces of the impeller vanes. Figure 5.11a and Figure 5.11b show the approximated Bézier curves for $r-\theta$ of the pressure and suction vanes, respectively. Figure 5.12a and Figure 5.12b show the approximated Bézier curves for $z-r$ of the pressure and suction vanes, respectively. In Figure 5.11 and Figure 5.12, the parameters used are $m=4$ and $n=2$. Thus, the corresponding size of the Bézier patch is 5×3.



(a) $r-\theta$ plot on pressure side. (b) $r-\theta$ plot on suction side

Figure 5.11: $r-\theta$ plot of impeller meridional plot and its fitted Bézier curve.



(a) $z-r$ plot on pressure side. (b) $z-r$ plot on suction side

Figure 5.12: $z-r$ plot of impeller meridional plot and its fitted Bézier curve.

The Bézier control points of the pressure and suction vanes share the same $z-$ and $r-$coordinates, in which some of them are held constant throughout the optimization process due to other physical constraints. The Bézier control points between the pressure and suction vanes vary most in $\theta-$coordinate, because the vane thickness is controlled through $r-\theta$ plot. The initial guess for Bézier control points is constructed from the original design of the impeller. In this study, the $z-$, $r-$, and $\theta-$coordinates of the Bézier control points are the design variables. For each functional evaluation, a set of Bézier control points are sampled, and a novel geometry of impeller is constructed, and evaluated using the predictive CFD impeller model.

5.5.2 3D CFD model for slurry pump impellers

In this section, we summarize the previous work of Pagalthivarthi et al. [210] in developing a CFD wear rate prediction for an industrial centrifugal slurry pump impeller. Figure 5.13 shows the 3D computational domain and boundary conditions of a centrifugal pump impeller in Cartesian coordinate system. Due to the angular symmetry and rotational invariance, only the region between pressure and suction sides of two vanes in the impeller are considered. The computational domain includes a region bounded by two vanes, two shrouds of the impeller, as well as the extensions downstream and upstream of the vanes. A set of governing equations is then derived based on 3D cylindrical coordinates with respect to a reference frame rotating with the impeller. Spalart-Allmaras model [212] is utilized to solve for the turbulent eddy viscosity. Comparison between $k-\varepsilon$ and Spalart-Allmaras turbulence models in 3D multi-size particulate flows is discussed in Pagalthivarthi et al [213]. The inlet velocity boundary condition is applied at the inlet surface B1. The stress free boundary condition is applied at the outlet surface B2. The blade surfaces B3 and B4, the hub surface B5, and the shroud surface B6 are treated as a wall, where Spalding wall functions [214] are utilized. On the surfaces B7, B8, B9, and B10, periodic boundary conditions are applied.

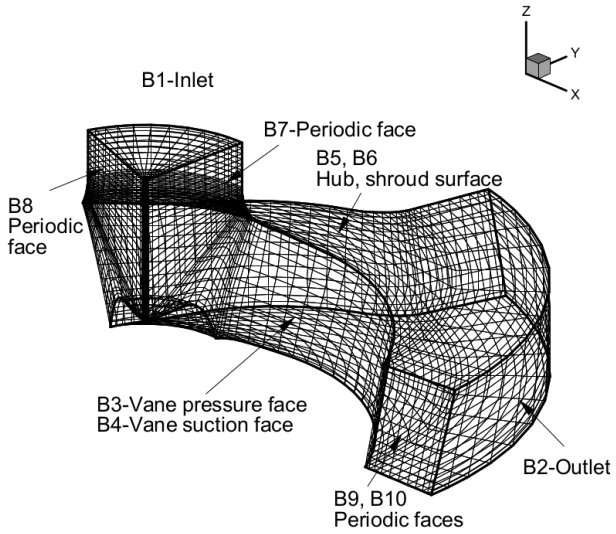


Figure 5.13: Three-dimensional pump impeller with mesh and its boundary conditions [210].

The Streamwise Upwind Petrov Galerkin [215] is utilized to formulate the finite element problem. The system of nonlinear equations for the mixture momentum, solids momentum, and solid concentration are iteratively solved with Newton's method where the under-relaxation factors are added to support the numerical convergence of the solutions. In the implementation, the system of algebraic equations are solved by Intel PARDISO solver [216], which is a shared memory parallel linear solver based on OpenMP. Only the solution of the linear system of equations is parallelized. All inner and outer iterations are repeated until the infinity norms of the absolute error, i.e. the difference between new solution and old solution, of all field variables reach values less than 10^{-5}.

After the CFD solution is obtained, a constitutive material model is utilized to predict the erosion wear rate [217]. The constitutive wear model is constructed empirically based on experimental measurements and approximates the total wear rate as the sum of impact and sliding wear rates. The impact wear rate is considered as a function of the particle size, impingement angle, local concentration, solids density, and velocity magnitude of the solid particles. The sliding wear rate is expressed as a function of the local concentration, solids shear stress, solids tangential velocity, and friction velocity. The wear coefficients are

determined experimentally by curve fitting with respect to experimental wear measurement for a specific wear-resistant alloy. The total wear rate normal to the surface, calculated in the units of $\mu m/hr$, gives the local erosion wear rate.

5.5.3 Feasibility classification

The unknown constraints in this case study are mainly from two sources. Both unknown constraints can only be assessed using the CFD erosion impeller wear model. The first type of unknown constraints is associated with the non-convergent behavior of the CFD impeller wear model. It is hypothesized that if the design is not plausible, then the CFD impeller wear model would not converge. The second type of unknown constraints occurs when the numerical solution is beyond the physical range. That means, given the design, the CFD numerical solver is ill-conditioned and has converged to an unstable solution.

Different from the unknown constraints, the known constraints can be fairly easy to assess once the inputs, i.e. the coordinates of Bézier control points are given. The known constraints are primarily physics-based validations of an impeller design, which are divided into two known constraints. The first constraint is that the impeller vane must have positive thickness. The second constraint is that the $r - \theta$ plot must be monotonically decreasing with respect to r. In the implementation, the second constraint is relaxed and not strictly enforced, as long as the deviation is within a user-defined tolerance.

To implement the unknown constraints, each simulation is attached to a wall-clock timer. If the predictive CFD wear model fails to converge to a solution, the design is considered infeasible in the classification GP. The timing threshold is chosen in such a way that a typical impeller wear simulation should converge, assuming a plausible design. For the second type of unknown constraints, a threshold wear rate is imposed. If the predicted wear rate is higher than this threshold, then the design is classified as infeasible in the classification GP.

For the known constraints, the feasibility checking function is implemented based on

the formulation of Bézier patch, and is embedded within the acquisition as the known constraint indicator function $I(\lambda(\boldsymbol{x}) \leq \boldsymbol{c})$ in Equation 5.7. Infeasible designs which fail the feasibility checking function are assigned zero value in the acquisition function.

5.5.4 Simulations and Results

In this case study, the degrees of Bézier surface patch are $m = 4$ and $n = 2$, resulting in 5×3 Bézier patch for the impeller pressure and suction vanes, respectively. The CFD simulation assumes a constant particle size (or mono-size) and thus the number of species in the particle size distribution is simplified to one. Considering all three $z-$, $r-$, and $\theta-$coordinates, a 33-dimensional input $\boldsymbol{x}$ is formed for each CFD simulation. The optimization procedure is carried out for pump assembly Z0534, at the input operating conditions of $Q = 89637.900$ gpm, $H = 50$ m, $N = 849.000$ RPM, $\eta = 82.400$, $d_{50} = 300\mu$m, $d_{85} = 690\mu$m, $d_{\text{eff}} = 495\mu$m, $C_v = 20\%$, and %BEPQ = 99.6%, where Q is the volumetric flow rate, N is the impeller angular speed, η is the hydraulic efficiency. d_{50}, d_{85} are the 50th and 85th percentile of the particle size distribution. $d_{\text{eff}} = 495\mu$m is the effective particle size, which is calculated as the average of the d_{50} and d_{85} and used as an input for mono-size species in the CFD simulation. $\%BEPQ$ is the percentage of best efficiency point flow rate. The design impeller diameter is 1.7018m, the shroud diameter is 1.7780m, the suction diameter is 0.6604m, and the discharge diameter is 0.6096m. The pump specific speed N_s in US units in 1425.6.

EI acquisition function is used to find the next sampling point in the acquisition hallucination batch. Gaussian and exponential kernels are used for the objective and classification GPs, respectively. A careful lower and upper bounds are chosen for the hyper-parameter of two GPs. The parameters $B_{\text{acquisition}}$, B_{explore}, and B_{classif} are set to 7, 5, 3, respectively. It indicates that at each optimization iteration, 15 CFD models are simulated concurrently in a parallel manner. Covariance matrix adaptation evolution strategy (CMA-ES) [144] is used as an auxiliary optimizer to find the location where the acquisition function is maximized.

A threshold wear rate of $1000\mu m/hr$ is imposed to classify the feasibility of the design. Also, a wall-clock timer of 7 hours is attached to each simulation, by which if the CFD model fails to converge, then the design is classified as infeasible. A fine mesh settings is used during the optimization process, such that the same mesh is used for all simulations.

For each simulation, a post-processing script is devised to extract the quantity of interest as the objective functional value, which is the average wear of the suction and pressure vanes of the impeller for a particular design. The implementation is performed on an Intel Xeon CPU E5-2637 v2 @3.50 GHz with 8 cores, 16 logical processors, and 128 GB RAM on Linux Ubuntu 16.04 platform. Fortran is used as a main language for CFD simulation, where parallelization within the CFD simulation is enabled via PARDISO solver as described above. MATLAB is used as the main programming language to implement the BO based on DACE toolbox [143] [218]. Python and Shell programming languages are used to develop the interface between the pBO-2GP-3B optimizer and the simulation, whereas Fortran with OpenMP, parallelized by the open-source PARDISO [216], is used as the main language for the CFD simulation. 480 initial random sampling points using Monte Carlo method are simulated concurrently to construct the initial GPs in $d = 33$ dimensional design space, where the original design and simulation is the first functional evaluation.

The optimization process is carried out for 89 optimization iterations, where each iteration corresponds to 15 parallel simulations. This results in a total of 1815 simulations of the 3D CFD impeller wear model. The average computational time to construct a batch of size 15 for each iteration is approximately 3-5 hours, depending on the settings of the auxiliary optimizer, which is CMA-ES in this case.

Figure 5.14 presents the convergence plot of pBO-2GP-3B for the design optimization of the slurry pump impeller within the feasible range. The infeasible designs are classified and assigned zero objective functional value to visualize. Technically, for infeasible designs, the objective functional values either do not exist because the simulations have failed to converge, or lie outside the physically plausible range, which is imposed before-

hand. Non-convergent simulations are scattered throughout the optimization process, as the optimizer explores unknown regions. All the best-so-far designs are found by the exploitation batch, as the goal of the exploitation batch is to improve previous designs.

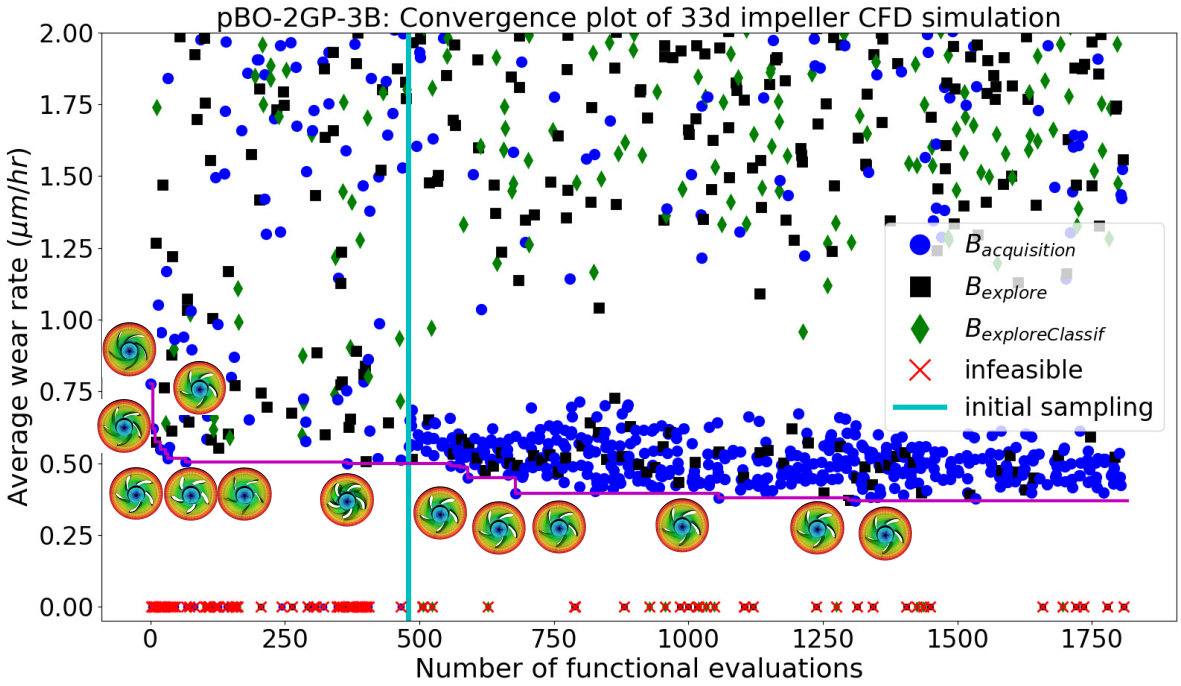
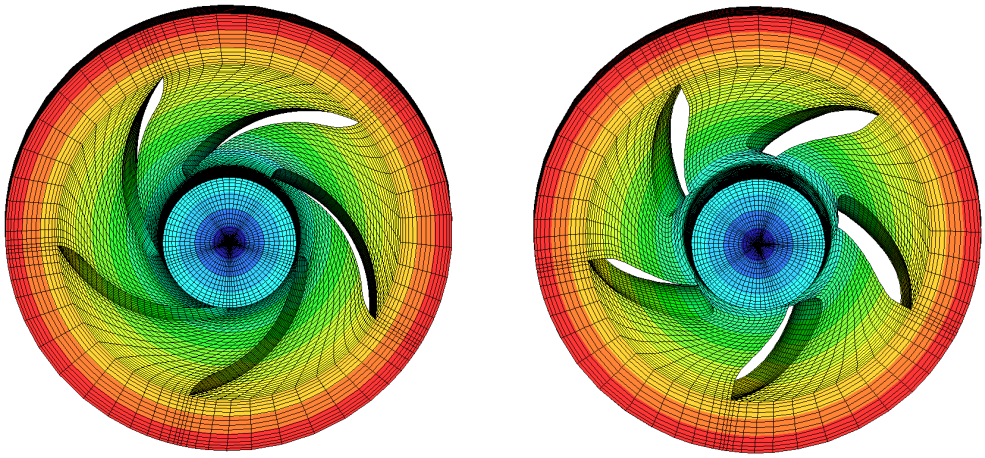


Figure 5.14: The convergence plot of pBO-2GP-3B for the impeller design optimization problem. The infeasible designs are assigned as zero for the objective GPs and are marked as crosses. The batch sizes are set as $B_{acquisition} = 7$, $B_{explore} = 5$, $B_{exploreClassif} = 3$. 15 CFD simulations are ran concurrently for each iteration.

The first functional evaluation denotes the original design, which is predicted at $0.7770 \mu m/hr$ for the wear performance. The first iteration starts counting at iteration 480, because there are 480 initial sampling points and the starting index is 0. The optimal design, which is evaluated 0.1527 $\mu m/hr$, is found at the iteration 536 in the first batch, at the fifth sampling point according to the hallucination scheme. The average suction side and pressure side wear on the vane, which is the quantity of interest and the objective functional value, is assessed through the multiphase CFD simulation. This value starts dropping from 0.7770 to the following values in the best-so-far solution, sequentially: 0.6203, 0.5732, 0.5485,

0.5316, 0.5181, 0.50141 0.4991, 0.4898, 0.4844, 0.4503, 0.3937, 0.3792, 0.3718, 0.3683 $\mu m/hr$.

Figure 5.15 presents two meshes, where Figure 5.15a depicts the original design, and Figure 5.15b depicts the optimal design of the impellers. The pressure and suction vane designs are structurally different because the Bézier control points are chosen as design inputs, which have been optimized against the average wear rate. In the optimal design, the vane thickness increases in a non-uniform manner, and achieves its maximum thickness near the trailing edge of the vane, before being tapered together at the trailing edge. It is found that the sweep angle in the optimal design is lower compared to the original design. As a result, the length of the vane decreases substantially. The twist angle increases and the outlet angle slightly increases in the optimal design, compared to the original design.



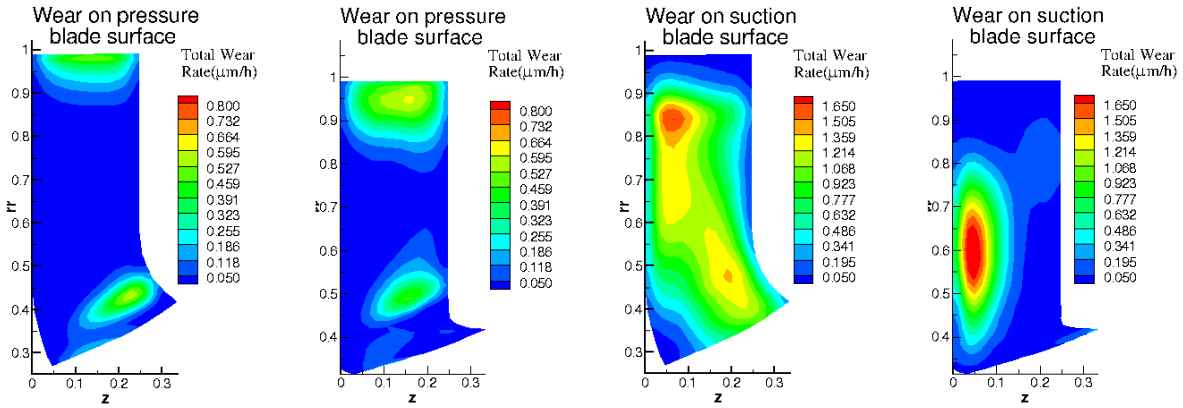
(a) Original design. (b) Optimal design.

Figure 5.15: Comparison between the CFD meshes of the original and optimal impeller designs.

Figure 5.16 presents the comparison of wear performance between the original and optimal vane designs. Figure 5.16a and Figure 5.16b compare the total wear plot in $z-r$ coordinates of the original and optimal impeller designs on the pressure side of the vanes. Figure 5.16c and Figure 5.16d compare the total wear plot in $z-r$ coordinates of the original and optimal impeller designs on the suction side of the vanes. The result indicates that in the original design, the suction side of the vane is associated with higher wear rates. By

optimizing and changing the flow pattern, the suction vane shows significant improvement in wear performance. While the average wear on the pressure vane is essentially similar in terms of magnitude, the average wear on the suction vane is significantly lower because the wear hot spot is localized. Furthermore, the magnitude of the wear hot spot is the same with the wear hot spot in the original design. The average wear rate reduces from $0.7770 \mu m/hr$ to $0.3683 \mu m/hr$. The optimal design reduces 51.70% of the average total wear compared to the original design. There is a possibility of the BEPQ being shifted as a result of the change in the design, which will be investigated in future studies.

The predicted hydraulic mixture efficiency of the optimal design is 92.49%, compared to 94.18% that of the original design, indicating 2% drop in predicted efficiency. The predicted head on slurry is 56.95 m for the optimal design, compared to 58.18 m for the original design. On the suction side of the vane in the original design, a highly localized maximum wear rate is predicted near the hub, near $rr = [0.8, 0.9]$ scaled region because of the local increase in the particle velocities, as shown in Figure 5.16c. In the optimal design, the flow is altered so that the wear region is localized on the suction vane with the same magnitude as shown in Figure 5.16d.



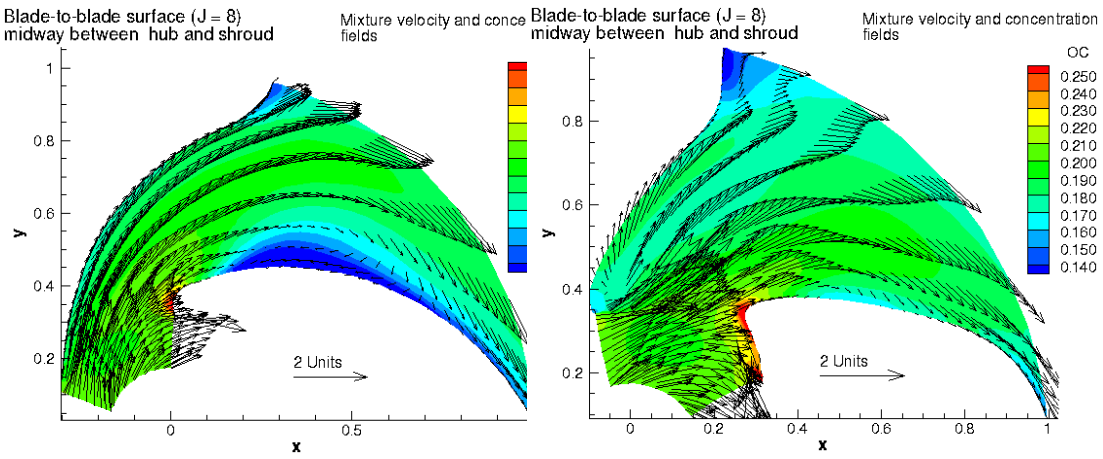
(a) Vane pressure side of original design

(b) Vane pressure side of optimal design

(c) Suction side of vane of original design

(d) Suction side of vane of optimal design

Figure 5.16: Comparison of total wear between the original and the optimal designs on both sides of the impeller vanes.



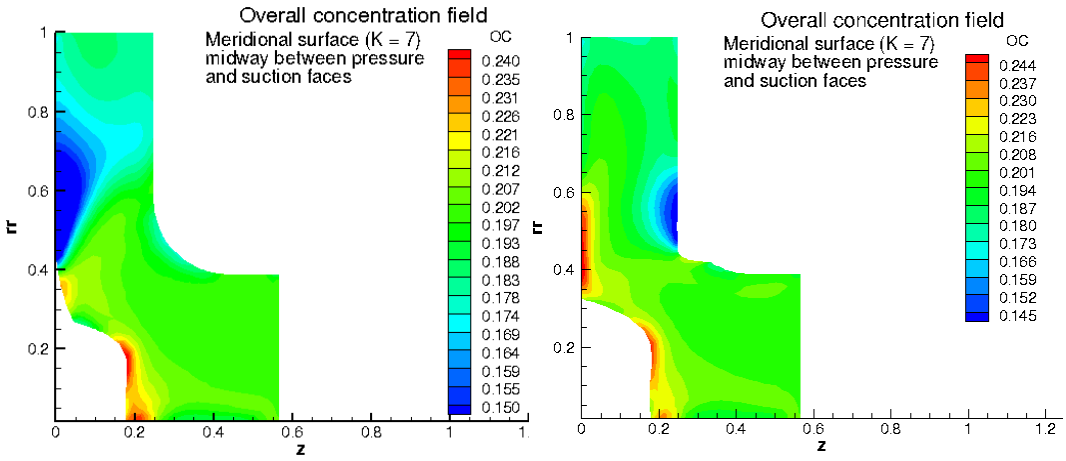
(a) Original design.

(b) Optimal design.

Figure 5.17: Comparison of mixture velocity and solids concentration between hub shroud and front shroud of the original and optimal impeller designs indicates that in the original design, the particles' maximum velocities occur near the suction vane and produce significant wear on the suction side. On the other hand, in the optimal design, the flow field pattern has been alternated such that the particles achieve its maximum velocities between the pressure and vane. Therefore, the total wear on the suction vane is reduced significantly.

Figure 5.17 shows the comparison of the mixture velocity and concentration fields on the hub shroud and front shroud of the original and optimal design. In the original design (Figure 5.17a), the particles accelerate within the impeller flow field and achieve the maximum velocity near the suction side of the vane, close to the trailing edge. In the optimal design (Figure 5.17b), the velocity field indicates higher velocities midway between the vanes.

Figure 5.18 shows the comparison of the overall concentration in a meridional surface of the original and optimal design in the cylindrical coordinate system, in Figure 5.18a and Figure 5.18b, respectively. In the original design (Figure 5.18a), the overall concentration is less uniform near the outlet and near the back shroud of the impeller, $rr = [0.4-0.8], z = [0.0 - 0.1]$. In the optimal design (Figure 5.18b), the overall concentration field indicates a more uniform behavior, although there is still a small region with low concentration near the front shroud at $rr = [0.45-0.60], z = 0.2$. Comparing the region with low concentration in the optimal design (Figure 5.18b) to that in the original design (Figure 5.18a), it is obvious



(a) Original design. (b) Optimal design.

Figure 5.18: Comparison of overall solids concentration between hub shroud and front shroud of the original and optimal impeller designs indicates the overall concentration is not uniform near the outlet in the original design. In contrast, in the optimal design, the overall concentration is more uniform near the outlet, although there are still some local regions with high concentrations within the impeller.

that the region is smaller and the concentration field is more uniform in the optimal design.

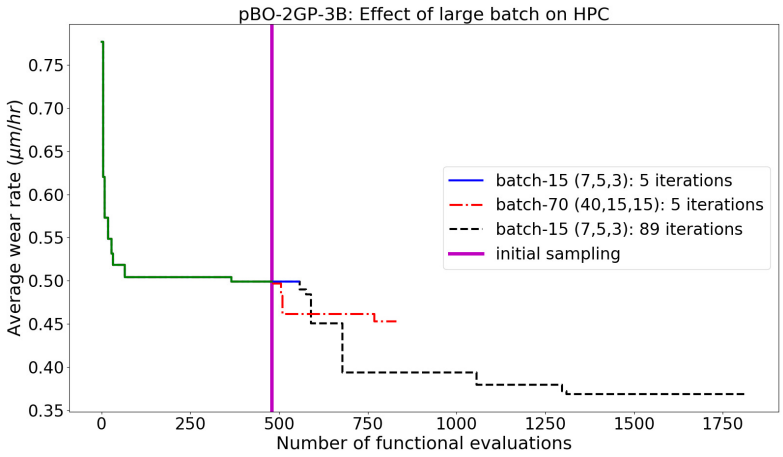


Figure 5.19: Comparison between different batch settings on the convergence plot shows a significant advantage with large batch size. The initial sampling process occurs up to 480 functional evaluation and is denoted by a solid vertical line. pBO-2GP-3B with batch-15 (7,5,3) stagnates with a solid (blue) line in 5 iterations, whereas pBO-2GP-3B batch-70 (40,15,15) make considerable improvement in 5 iterations.

Figure 5.19 presents a comparison between two different batch settings, batch-15 and batch-70, after 5 iterations performance. The former one is associated with a total batch size of 15, where the first, second, and third batch sizes are 7, 5, 3, respectively. The later

one is associated with a total batch size of 70, where the first, second, and third batch sizes are 40, 15, 15, respectively. The significant improvement of batch-70 over batch-15 in 5 iterations, as shown in Figure 5.19, demonstrates the efficiency of the proposed pBO-2GP-3B method in HPC environment. It agrees with the intuition that larger batch is more effective in solving the parallel optimization problem, where the batch size limit depends on the size of the HPC. However, we also note that there is a computational bottleneck in the batch construction process, as the sampling point is selected sequentially within a batch, which may reduce the actual efficiency of the proposed method in practical settings.

5.6 Discussion

Even though several advanced features have been included, pBO-2GP-3B also has some limitations. One of the drawbacks of the current approach pBO-2GP-3B is the scalability of the algorithms, which suffers in both GPs, as the number of data increases more than 10,000. To cope with the scalability of GPs, a possible solution is to decompose a large dataset into smaller ones and formulate as distributed GPs and weighted linear prediction as in our previous work [1]. Another drawback is the lack of an asynchronous feature, which means that the batch of simulations must be finished, before the optimization process can move on to the next batch. In the CFD engineering example, this idea is implemented by enforcing a maximum time running for of the CFD simulation. The third drawback of the current approach is the sequential nature of the sampling points within a batch. This is particularly important for a HPC, for example, with hundreds to thousands processors available, because the current approach would take more time to construct a batch than simulate it. This begs for further development for a method which searches the sampling points concurrently, as proposed by Daxberger and Low [206], or an adaptive approach that does not wait until a batch is finished, but rather simulates as the inputs are ready. We also note that too large batch would indeed lead to low optimization efficiency because the limit scale of GPs would be reached in less wall-clock time, yet both GPs do not have enough

batch to sequentially control the location of next sampling points. The last drawback of the proposed pBO-2GP-3B method is the simple exploring scheme that maximizes the posterior variance of the objective GP in the second batch B_{explore}, which tends to sample on the border of the domain. The issue is more profound in high-dimensional problem. This can be corrected by more elaborate variance-based sampling scheme, such as Integrated Mean Squared Error (IMSE) [219, 220, 221].

The development time is minimized and thus suitable for fast deployment in industrial applications. Particularly, the development of classification GP for blind constraints is user-friendly, and thus attractive for engineers and designers, who are typically the users of the optimization package in engineering simulation-based problems. Known constraints which are ignored by some users can conveniently become unknown constraints. Such actions reduce the development time for code deployment, yet does not severely affect the functionality of the algorithm, and thus pBO-2GP-3B can be considered as user-friendly. Implementing pBO-2GP-3B in HPC where a job scheduler is available is even easier, since once the design is readily available, one can simply submit the job, and search for the next sampling point while the simulation is in the submission queue.

One remarkable feature of pBO-2GP-3B is the adaptive interpolation of the infeasible data in the objective GP. After each iteration, the objective GP is first reconstructed using only the feasible data points in the dataset, then updated later, assuming the posterior mean as actual observations. The main purpose of this interpolation process is to truly estimate the variance $\sigma^2_{\text{objective}}$ for uncertainty quantification purposes. This is because the computation of the variance $\sigma^2_{\text{objective}}$ only depends on all the locations $\boldsymbol{x}$'s of the dataset and its covariance model, instead of the actual observation y's. Thus by hallucinating the infeasible data points, the variance $\sigma^2_{\text{objective}}$ is accurately quantified. Thus the goal of two pure exploration batches in Algorithm 3 is assured. The interpolation scheme sometimes leads to false optimum in infeasible region. However, the acquisition function is modified in such a way that the binary GP classifier would penalize more in the acquisition function,

and thus pBO-2GP-3B would move away from infeasible region.

A critical point in using pBO-2GP-3B algorithm is that the wall-clock time is not simply just the time to perform parallel simulations, but also include others, such as queue time on the HPC if the computational resource is shared, time to construct different batches within one iteration, time to download and upload data from different sources, etc. Particularly, larger batch size takes longer time to prepare an input to evaluate. If that amount of time is somewhat comparable to the simulation time, there is a combination of batch sizes which yields optimal wall-clock time performance, but the exact answer depends on the computational time of a specific simulation. Therefore, in practice, one is only beneficial from increasing the batch size if the asynchronous feature is enabled. This point, again, refers back to the co-optimize to find the sample points within a batch concurrently.

Because of the nature of binary classification problem in the classification GP, the algorithm pBO-2GP-3B performs well if the initial sampling dataset contains at least one feasible and one infeasible data points. Otherwise, the classification GP is not well calibrated, and thus would not yield a good prediction at the beginning phase of optimization process. However, as the optimization advances, the classification GP should converge relatively fast because of the imposed linear covariance kernel, and thus the estimation of the acquisition function is more accurate once the performance of the classification GP improves.

There is a considerable difference between acquisition function in Equation 5.7 and Equation (4.2) in Schonlau et al [174]. First, in Schonlau et al. [174], the constraints are unknown and quantifiable, in the sense that the objective GP returns a real value for the model, from which the probability of satisfying the constraints can be estimated. Schonlau et al. [174] method does not handle the case where the functional evaluator crashes, or does not return any response. For our test problems, if the response does not exist in the infeasible domain, the Schonlau et al. [174] method is not applicable. Should an artificial value be used, as in the constant liar heuristic in Ginsbourger et al. [187], the objective

GP model would changes sequentially to reflect the constant liar heuristic. As a result, the objective GP model would fail to predict the boundary between infeasible and feasible region, because of the underlying smoothness assumption of the GP formulation. Indeed, it is one of the challenges in engineering domain, where the response does not exist. Second, Equation 5.7 in our chapter handles this problem by borrowing another binary classifier, which classifies whether the domain is feasible or infeasible. Introducing a probabilistic classifier allows the discontinuity between the feasible and infeasible regions, because the objective GP model and the classifier model are independent from each other. A taxonomy of constrained optimization problems is discussed in Digabel and Wild [106] for classifying constrained optimization problems. In this work, the GP is chosen to be the probabilistic classifier, with exponential kernel, instead of Gaussian kernel.

Indeed, one of the most important aspects in solving simulation-based engineering optimization problem is the imbalance dataset, because the classification GP is still functioning as a binary classifier. Too many divergent cases would deteriorate the performance of the classification GP and exhaust the optimization algorithm. Furthermore, since pBO-2GP-3B is a GP-based approach, it also suffers from the scalability. Thus, design the support domain for input variables such that there would be more convergent simulations is recommended. This action can be achieved through examining the results of the initial sampling cases.

There are two ways to obtain a better optimization results. The first one is to expand the support intervals that defines the lower and upper bounds for the input. As the design space grows larger, the global optimum result is guaranteed to be better. However, for simulations that are sensitive to input design variables, such as CFD, this would lead to more divergent cases and exhaust the optimization algorithm, as discussed above. The second one is to increase the number of inputs or degrees of freedom. For example, in the engineering CFD example above, one could also increase m and n parameters, which control the size of the Bézier patch to obtain a better result. This direction is impacted

by the curse-of-dimensionality, and thus one should be cautious regarding the trade-off between the approximation error and the curse-of-dimensionality.

Since the method can be thought as a natural extension of GP-BUCB [199] and GP-UCB-PE [200], assuming the batch size of the third batch, the pure exploration for the classification GP, is zero, i.e. $B_{\text{classif}} = 0$, the cumulative regrets is thus bounded by the maximum upper bound of these methods, which is GP-BUCB because GP-UCB-PE has been proved to be better than GP-UCB by a ratio of $\sqrt{B_{\text{acquisition}} + B_{\text{explore}}}$. Indeed, the pure exploration batch of the classification GP plays an important role in preventing convergence to local minima when the infeasible space dominates the high-dimensional input space. While the acquisition function is designed in such a way that the algorithm only samples at feasible regions, the pure exploration batch of the classification GP forces the algorithm to also explore other uncertain regions, with the hope to find some missing feasible regions that have not been found previously. The careful theoretical convergence analysis study remains unsolved and open for future work.

A GP binary classifier is competitive in high-dimensional space binary classification problem with relatively fast convergence rate. Many other binary classifiers, including kNN [181], AdaBoost [182], RandomForest [183], support vector machine [184], and least squares support vector machine [185], are implemented to compare the numerical performance to the proposed pBO-2GP-BO method. However, all of them are cursed by the high-dimensionality of the optimization problems. The GP classifier is an excellent choice of classifier for two reasons. First, the uncertainty of the GP classifier is naturally quantified by the posterior variance σ^2. Second, because of the uncertainty quantified, the GP classifier is then forced to learn at the most unknown regions. We note that there is not so many binary or multi-class classifiers have uncertainty quantification feature in the context of machine learning.

It is noted that the comparison study is rather limited, since the comparison study only concerns with the convergence as a function of iterations, yet many other norms are also

equivalently important. One of the most realistic norms is the actual computational runtime, i.e. wall-clock time, which can be used to measure the efficiency of different optimization algorithms. There are many factors which can alternate the comparison results, including computational platforms, operating systems, hardware, implementation, as well as objective functions, initial designs, constraints, random number generators. To truly compare one optimization algorithm with others, a large scale benchmark study is needed, and thus it is left to future work.

The proposed pBO-2GP-3B method can be easily extended to solve for equality constraints problem. For example, the objective function can be modified to incorporate the equality constraints to a penalized objective function, in which the penalty is applied directly on the unsatisfied equality constraints, as in Picheny et al [176]. GP formulation accounts for some intrinsic noise by assuming that the observations are jointly Gaussian [141]. However, for complex noisy problems where the model response is stochastic, rather than deterministic, a more advanced approach is needed, such as stochastic kriging [222]. The model responses could be the objective functional evaluator, as well as the feasibility condition. This problem remains unsolved and is left to future work.

In the previous numerical and engineering examples, the batch sizes are held constant throughout the optimization process. However, it can be adaptively tuned based on a fixed computational budget. For example, more computational efforts can be focused on reducing the variance σ^2 in the first phase until σ^2 hits a critical number, then more computational efforts can be spent on the acquisition hallucination batch in the second phase for exploitation purpose. This opens up another research question for the dynamic computational resource allocation in the context of HPC, which is a NP-hard problem and will be addressed in future work. Another possible extension is the asynchronous BO, where if a simulation is finished, then a new simulation is readily dispatched.

5.7 Summary

In this chapter, we present a novel BO method pBO-2GP-3B, which is aimed at computationally expensive and high-fidelity engineering optimization problems. Two GPs are utilized: the objective GP for the objective function, and the classification GP for the probabilistic classification. The proposed pBO-2GP-3B significantly brings down the computational cost, based on the premise of available HPC resources. By massively parallelizing the simulation, i.e. each processor handles a single simulation, the diminishing return described in Amdahl's law is avoided, representing a better use for computational resources, which in turn further reduces the wall-clock time.

Additionally, pBO-2GP-3B supports both known and unknown constraints in simulation-based engineering optimization problems, in which the known constraints are defined beforehand, and the unknown constraints are only known once the simulations have been ran. The known constraints are penalized directly into the acquisition using a constraint indicator function, which assigns zero value if the known constraints are violated. On the other hand, the unknown constraints are learned through the classification GP, and the predicted feasibility is then coupled to the acquisition function, using a simple product rule.

The proposed method is demonstrated using a 2D three-hump camel, 2D Rastrigin function, and 6D Rastrigin functions, showing good convergence rate for both numerical problems. It is then applied to a real-world engineering problem for the design optimization of a slurry pump impeller. The predicted result indicates 52.60% reduction in average wear performance, and denotes the potential application of the proposed method in HPC systems.

CHAPTER 6
CONCLUSION

6.1 Discussion

In this dissertation, we are concerned with UQ, UP, and optimization under uncertainty problems in the context of computational materials science, where ICME tools are utilized to establish the PSP linkage. Since the uncertainty is a critical element in ICME tools and microstructure, the UQ toolbox is used in concert with ICME tools to enhance the predictive capability. Both physics-based and data-driven approaches are adopted to solve the UQ problems in computational materials science.

The contribution of this thesis is to advance the current knowledge of ICME tools in several frontiers. In Chapter 3, we developed a novel method to lift the time scale limit of the ICME tools and demonstrate its usefulness in microstructural evolution. In Chapter 5 and 6, two other novel optimization methods are proposed to solve for the inverse problem, where the computationally expensive and complex ICME tools are considered as the functional evaluators.

In Chapter 3, the time scale limit of nanoscale ICME tools is alleviated using the proposed stochastic ROM model. The stochastic ROM model is particularly useful for high-fidelity nanoscale ICME tools, such as MD simulations and DFT calculations. The computational burden in MD simulations to predict the microscopic QoIs can be significantly reduced using the stochastic ROM, where the coefficients are trained based on the QoIs' history. Furthermore, because the computational cost to propagate the stochastic ROM is substantially cheaper, e.g. the cost to solve a 1D PDE is much smaller than to solve for $\mathcal{O}(10^6)$ ODEs, the microscopic QoIs can be conveniently propagated using the stochastic ROM. In the demonstration, we used the chord-length distribution and the grain area as the

microstructural descriptors to demonstrate the effectiveness in mesoscale ICME tools. For the nanoscale MD simulations, several macroscopic QoIs are considered as the statistical descriptors of the simulation cells.

In Chapter 4, we employed SG, PCE, and SC methods to investigate the dendritic growth in Al-4wt%Cu binary alloys, with respect to thermodynamic and process parameters. It is shown that the dendritic morphology is affected by process parameters and thermodynamic parameters, which are a part of constitutive models. The employment of SG in high-dimensional problems is to relieve the computational burden in exploring the high-dimensional input space. The uncertainty of dendritic growth is quantified using the SG approach, in which PCE is utilized to represent the QoIs as polynomial approximations, and SG is employed to interpolate and integrate the QoIs on high-dimensional space.

In Chapters 5 and 6, we presented two extensions to the BO method to solve for the inverse problem in high-dimensional space. It is a part of our effort in extending the classical BO method toward a ”versatile” BO method. BO is a powerful optimization method in high-dimensional problem. The significant advantages of BO is data-driven, adaptive, UQ, derivative-free, and easy to extend. The theoretical foundation of BO has been half a century, yet the method is still being further developed and extended to scope with modern-era problems.

In Chapter 5, the batch-parallel extension is crucial in reducing the computational runtime for the inverse high-dimensional problem with high-fidelity ICME tools through increased parallelism on HPC. The effectiveness of the proposed algorithm is verified through the usage of HPC with larger batch. In other words, increased effectiveness is observed to correlate with the size of the batch, and subsequently the size of the HPC. It is noteworthy that with the proposed technique, the usage of high-fidelity ICME tools as functional evaluators are computationally possible and tractable with the aid of HPC.

Chapter 6 describes another important extension of BO method to solve for the mixed-integer optimization through categorical regression decomposition approach. The theory

is verified mathematically, as well as empirically through comparison with genetic algorithms. In computational materials science, there are many situations in which the continuous variables simply cannot describe the discrete nature of the inputs. For example, there are several sheet metal related manufacturing processes, including deep drawing, shallowing drawing, bar drawing, wire drawing, and tube drawing processes. For extrusions, there are hot extrusion, cold extrusion, warm extrusion, friction extrusion and micro-extrusion processes. Without the discrete variables, it would be difficult to represent process parameters.

Non-intrusive UQ methods are concerned in this dissertation. It is noteworthy that intrusive UQ methods integration to ICME tools have been utilized in previous work, where interval analysis and generalized intervals are invoked to solve the model-form UQ problem in MD simulations [223, 224, 225, 226]. Distributed local GP approach is also proposed to search for the minimum energy path in high-dimensional complex potential energy surface [1]. It has been shown that the local GP approach is scalable and favorable to large dataset. Preliminary result on local GP approach also indicates that the GP framework is scalable to $\mathcal{O}(10^6)$ data points.

6.2 Summary of the Dissertation

The ultimate goal of the dissertation is to advance the UQ methodologies in the context of computational materials science. In this thesis, we are concerned with UQ, UP, and optimization under uncertainty in ICME context. Two methods are proposed in the first half of the thesis to quantify and propagate the uncertainty, accompanying by MD, PF, and kMC materials engineering examples. Two other optimization methods based on Bayesian optimization approach are adopted and extended to parallel and mixed-integer directions, respectively. They are demonstrated by several design optimization examples.

In Chapter 3, we proposed to propagate uncertainty in microstructural evolution, using kMC, PF, and MD models. The methodology relies on a stochastic ROM based on Fokker-

Planck equation, where the ROM parameters are trained using ML approaches. After training, the ROM is then used to propagate the uncertainty in the upper time scale beyond the classical time scale limit of the underlying ICME model. In the kMC example, the grain area in grain growth simulation is used as the statistical descriptor of the microstructure. In the PF example, the chord-length distribution is used as the statistical descriptor in the Fe-Cr decomposition, using Cahn-Hilliard equation as the govern equation. In the MD example, several macroscopic QoIs are used as the descriptors for the whole simulation cell of the system.

Chapter 4 is concerned with quantifying uncertainty in PF model, particularly with the dendrite morphology in Al-4%wt Cu binary alloys. The process-structure relationship is investigated. Several QoIs are proposed to quantified the dendrite morphology, using image processing techniques. The dendrite morphology are plotted on sparse grid, showing a clear relationship between processing and microstructure. The dendritic growth is affected by both processing and thermodynamic parameters, both quantitatively and qualitatively. Some parameters promote the growth of the dendritic secondary arm, whereas some other parameters promote the growth of the whole dendrite.

In Chapter 5, we extended the classical sequential BO method to solve known/unknown constrained optimization problems in a parallel manner. The parallel extension is aimed toward reducing the computational runtime through increased parallelism in HPC infrastructure. The extension is constructed by building two GPs, where the first GP depends on the objective function, whereas the second GP classifies the feasibility of the sampling locations. The method is then demonstrated using CFD example, where the typical runtime is 4-7 hours. The effectiveness of the methodology is shown to correlate with the size of HPC used to optimize the problem.

In Chapter 6, we extended the classical sequential BO method to solve a mixed-integer optimization problems in a sequential manner. The approach is constructed based on the decomposition scheme to break a large dataset to smaller datasets, and combine the posterior

predictions using a weighted linear average approach. The methodology is demonstrated using a design optimization example for mechanical and auxetic metamaterials.

6.3 Contributions

The novel contributions of the dissertation are highlighted as follows.

- A stochastic ROM is developed to propagate uncertainty in ICME models, where the time scale of the ROM is much longer than that of the underlying ICME models. The proposed ROM is based on Fokker-Planck equations, where parameters are trained using a collection of data from the ICME models. The time scale used in the ROM is generally larger than the ones in the ICME models, and the computational cost to solve for the stochastic ROM is significantly lower than the ICME models. This allows the uncertainty to propagate statistically in a faster time scale. A possible application of the proposed method can be developed to construct the simulation domain using the propagated microstructural descriptors.

- A UQ method is applied to to assess the sensitivities in the dendritic growth in Al-4%wt Cu. The dendritic morphology is shown to vary with respect to different parameters used in the PF model.

- A synchronous parallel BO method is proposed to optimize high-fidelity engineering simulations, with known and unknown constraints. The unknown constraint can scope with unexpected termination of the model, diverging solver, ill-conditioned simulations, with respect to varying input parameters. The proposed method is demonstrated on HPC.

- A sequential mixed-integer BO method is proposed to solve for a mixed-integer constrained optimization problem, where the constraints are known *a priori*. The constraints are penalized in the acquisition function of the BO methods, whereas the

discrete variables are used to decompose a large dataset to smaller ones. The *posteriori* predictions are then combined using a weighted linear average scheme, where the weights change adaptively according to the dataset. The proposed algorithm is shown to outperform the state-of-the-art genetic algorithms with different settings.

6.4 Future work

As the field of materials informatics advances, one seeks to combine physics-based and data-driven approaches to solve high-dimensional complex problems, with limited data. The data can be either obtained from experiment or from simulation. Multi-fidelity GP is a promising approach to combine data from different sources in computational materials science. The future work can include

- extended optimization under uncertainty methodologies toward a ”versatile” BO algorithm, which can solve multi-fidelity, multi-objective, scalable optimization problems on the HPC,
- applications of the proposed UQ methods to other ICME tools at continuum and quantum scales, such as crystal plasticity models and density functional theory calculations for specific types of materials,
- development of data-driven approaches to solve a multi-scale ICME problems, where cross-scale information integration is involved.

Appendices

REFERENCES

[1] A. Tran, L. He, and Y. Wang, "An efficient first-principles saddle point searching method based on distributed kriging metamodels," *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part B: Mechanical Engineering*, vol. 4, no. 1, p. 011 006, 2018.

[2] R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu, and W. Chen, "Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques," *Progress in Materials Science*, 2018.

[3] M. Groeber, S. Ghosh, M. D. Uchic, and D. M. Dimiduk, "A framework for automated analysis and simulation of 3D polycrystalline microstructures. Part 1: Statistical characterization," *Acta Materialia*, vol. 56, no. 6, pp. 1257–1273, 2008.

[4] A. Agrawal and A. Choudhary, "Perspective: Materials informatics and big data: Realization of the fourth paradigmi of science in materials science," *APL Materials*, vol. 4, no. 5, p. 053 208, 2016.

[5] M Grigoriu, "Reduced order models for random functions. application to stochastic problems," *Applied Mathematical Modelling*, vol. 33, no. 1, pp. 161–175, 2009.

[6] S. Sarkar, J. E. Warner, W. Aquino, and M. D. Grigoriu, "Stochastic reduced order models for uncertainty quantification of intergranular corrosion rates," *Corrosion Science*, vol. 80, pp. 257–268, 2014.

[7] M. P. Mignolet and C. Soize, "Stochastic reduced order models for uncertain geometrically nonlinear dynamical systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 197, no. 45-48, pp. 3951–3963, 2008.

[8] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[9] S. Siegert, R Friedrich, and J Peinke, "Analysis of data sets of stochastic systems," *arXiv preprint cond-mat/9803250*, 1998.

[10] R. Friedrich, C. Renner, M. Siefert, and J. Peinke, "Comment on Indispensable finite time corrections for Fokker-Planck equations from time series datai," *Physical Review Letters*, vol. 89, no. 14, p. 149 401, 2002.

[11] M. Siefert, A. Kittel, R. Friedrich, and J. Peinke, "On a quantitative method to analyze dynamical and measurement noise," *EPL (Europhysics Letters)*, vol. 61, no. 4, p. 466, 2003.

[12] J. Gottschall and J. Peinke, "On the definition and handling of different drift and diffusion estimates," *New Journal of Physics*, vol. 10, no. 8, p. 083 034, 2008.

[13] P. Sura and J. Barsugli, "A note on estimating drift and diffusion parameters from timeseries," *Physics Letters A*, vol. 305, no. 5, pp. 304–311, 2002.

[14] M. Ragwitz and H. Kantz, "Indispensable finite time corrections for Fokker-Planck equations from time series data," *Physical Review Letters*, vol. 87, no. 25, p. 254 501, 2001.

[15] M. J. Gander, "50 years of time parallel time integration," in *Multiple Shooting and Time Domain Decomposition Methods*, Springer, 2015, pp. 69–113.

[16] J. Nievergelt, "Parallel methods for integrating ordinary differential equations," *Communications of the ACM*, vol. 7, no. 12, pp. 731–733, 1964.

[17] P. Chartier and B. Philippe, "A parallel shooting technique for solving dissipative ODE's," *Computing*, vol. 51, no. 3-4, pp. 209–236, 1993.

[18] P. Saha, J. Stadel, and S. Tremaine, "A parallel integration method for solar system dynamics," *arXiv preprint astro-ph/9605016*, 1996.

[19] Y. Maday and G. Turinici, "A parareal in time procedure for the control of partial differential equations," *Comptes Rendus Mathematique*, vol. 335, no. 4, pp. 387–392, 2002.

[20] G. Bal, "On the convergence and the stability of the parareal algorithm to solve partial differential equations," in *Domain decomposition methods in science and engineering*, Springer, 2005, pp. 425–432.

[21] E. Lelarasmee, *The waveform relaxation method for time domain analysis of large scale integrated circuits: Theory and applications*. Electronics Research Laboratory, College of Engineering, University of California, 1982.

[22] M. J. Gander, "Overlapping Schwarz for linear and nonlinear parabolic problems," 1996.

[23] M. J. Gander, L. Halpern, and F. Nataf, "Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation," 1999.

[24] W. Hackbusch, "Parabolic multi-grid methods," in *Proc. of the sixth int'l. symposium on Computing methods in applied sciences and engineering, VI*, North-Holland Publishing Co., 1985, pp. 189–197.

[25] C. Lubich and A Ostermann, "Multi-grid dynamic iteration for parabolic equations," *BIT Numerical Mathematics*, vol. 27, no. 2, pp. 216–234, 1987.

[26] G. Horton and S. Vandewalle, "A space-time multigrid method for parabolic partial differential equations," *SIAM Journal on Scientific Computing*, vol. 16, no. 4, pp. 848–864, 1995.

[27] M. Emmett, M. L. Minion, *et al.*, "Toward an efficient parallel in time method for partial differential equations," *Communications in Applied Mathematics and Computational Science*, vol. 7, no. 1, pp. 105–132, 2012.

[28] M. Neumüller, "Space-time methods: Fast solvers and applications," PhD thesis, University of Graz, 2013.

[29] W. L. Miranker and W. Liniger, "Parallel methods for the numerical integration of ordinary differential equations," *Mathematics of Computation*, vol. 21, no. 99, pp. 303–320, 1967.

[30] A. Axelsson and J. G. Verwer, "Boundary value techniques for initial value problems in ordinary differential equations," *Mathematics of Computation*, vol. 45, no. 171, pp. 153–171, 1985.

[31] D. E. Womble, "A time-stepping algorithm for parallel computers," *SIAM Journal on Scientific and Statistical Computing*, vol. 11, no. 5, pp. 824–837, 1990.

[32] P. Worley, "Parallelizing across time when solving time-dependent partial differential equations," in *Proc. 5th SIAM Conf. on Parallel Processing for Scientific Computing, D. Sorensen, ed., SIAM*, 1991.

[33] D. Sheen, I. H. Sloan, and V. Thomée, "A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature," *IMA Journal of Numerical Analysis*, vol. 23, no. 2, pp. 269–299, 2003.

[34] Y. Maday and E. M. Rønquist, "Parallelization in time through tensor-product space–time solvers," *Comptes Rendus Mathematique*, vol. 346, no. 1-2, pp. 113–118, 2008.

[35] A. J. Christlieb, C. B. Macdonald, and B. W. Ong, "Parallel high-order integrators," *SIAM Journal on Scientific Computing*, vol. 32, no. 2, pp. 818–835, 2010.

[36] S. Güttel, "A parallel overlapping time-domain decomposition method for odes," in *Domain decomposition methods in science and engineering XX*, Springer, 2013, pp. 459–466.

[37] H Risken, *The Fokker Planck equation, Methods of solution and application 2nd Ed*. Springer Verlag, Berlin, Heidelberg, 1989.

[38] C. Honisch and R. Friedrich, "Estimation of Kramers-Moyal coefficients at low sampling rates," *Physical Review E*, vol. 83, no. 6, p. 066 701, 2011.

[39] G. Casella and R. L. Berger, *Statistical inference*. Duxbury Pacific Grove, CA, 2002, vol. 2.

[40] J. J. Stickel, "Data smoothing and numerical differentiation by a regularization method," *Computers & chemical engineering*, vol. 34, no. 4, pp. 467–475, 2010.

[41] H. Hassan, A. Mohamad, and G. Atteia, "An algorithm for the finite difference approximation of derivatives with arbitrary degree and order of accuracy," *Journal of Computational and Applied Mathematics*, vol. 236, no. 10, pp. 2622–2631, 2012.

[42] S. Wojtkiewicz and L. Bergman, "Numerical solution of high dimensional Fokker-Planck equations," in *8th ASCE Specialty Conference on Probablistic Mechanics and Structural Reliability, Notre Dame, IN, USA*, 2000.

[43] E. R. Homer, V. Tikare, and E. A. Holm, "Hybrid Potts-phase field model for coupled microstructural–compositional evolution," *Computational Materials Science*, vol. 69, pp. 414–423, 2013.

[44] S Plimpton, A Thompson, and A Slepoy, *Spparks kinetic monte carlo simulator*, 2012.

[45] A. W. Bowman and A. Azzalini, *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*. OUP Oxford, 1997, vol. 18.

[46] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandie, "MOOSE: A parallel computational framework for coupled systems of nonlinear equations," *Nuclear Engineering and Design*, vol. 239, no. 10, pp. 1768–1778, 2009.

[47] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics*, vol. 117, no. 1, pp. 1–19, 1995.

[48] A. McGaughey and M Kaviany, "Thermal conductivity decomposition and analysis using molecular dynamics simulations. part I. Lennard-Jones argon," *International Journal of Heat and Mass Transfer*, vol. 47, no. 8, pp. 1783–1798, 2004.

[49] P Borgelt, C Hoheisel, and G Stell, "Exact molecular dynamics and kinetic theory results for thermal transport coefficients of the Lennard-Jones argon fluid in a wide region of states," *Physical Review A*, vol. 42, no. 2, p. 789, 1990.

[50] A Dawid and Z Gburski, "Interaction-induced light scattering in Lennard-Jones argon clusters: Computer simulations," *Physical Review A*, vol. 56, no. 4, p. 3294, 1997.

[51] K. Laasonen, S. Wonczak, R. Strey, and A. Laaksonen, "Molecular dynamics simulations of gas–liquid nucleation of Lennard-Jones fluid," *The Journal of Chemical Physics*, vol. 113, no. 21, pp. 9741–9747, 2000.

[52] D. Reith and F. Müller-Plathe, "On the nature of thermal diffusion in binary Lennard-Jones liquids," *The Journal of Chemical Physics*, vol. 112, no. 5, pp. 2436–2443, 2000.

[53] M. Griebel, S. Knapek, and G. Zumbusch, *Numerical simulation in molecular dynamics: numerics, algorithms, parallelization, applications*. Springer Science & Business Media, 2007, vol. 5.

[54] B. Dünweg and W. Paul, "Brownian dynamics simulations without Gaussian random numbers," *International Journal of Modern Physics C*, vol. 2, no. 03, pp. 817–827, 1991.

[55] S Torquato, "Statistical description of microstructures," *Annual review of materials research*, vol. 32, no. 1, pp. 77–111, 2002.

[56] M. Groeber, S. Ghosh, M. D. Uchic, and D. M. Dimiduk, "A framework for automated analysis and simulation of 3D polycrystalline microstructures. Part 2: Synthetic structure generation," *Acta Materialia*, vol. 56, no. 6, pp. 1274–1287, 2008.

[57] Y. Liu, M. S. Greene, W. Chen, D. A. Dikin, and W. K. Liu, "Computational microstructure characterization and reconstruction for stochastic multiscale material design," *Computer-Aided Design*, vol. 45, no. 1, pp. 65–76, 2013.

[58] V John, I Angelov, A. Öncül, and D Thévenin, "Techniques for the reconstruction of a distribution from a finite number of its moments," *Chemical Engineering Science*, vol. 62, no. 11, pp. 2890–2904, 2007.

[59] I Steinbach, B Böttger, J Eiken, N Warnken, and S. Fries, "CALPHAD and phase-field modeling: A successful liaison," *Journal of phase equilibria and diffusion*, vol. 28, no. 1, pp. 101–106, 2007.

[60] U. R. Kattner, "The CALPHAD method and its role in material and process development," *Tecnologia em metalurgia, materiais e mineracao*, vol. 13, no. 1, p. 3, 2016.

[61] C. Marker, A. Ross, and Z.-K. Liu, "Design of materials processing using computational thermodynamics," in *Computational Materials System Design*, Springer, 2018, pp. 27–45.

[62] C. E. Campbell, U. R. Kattner, and Z.-K. Liu, "The development of phase-based property data using the CALPHAD method and infrastructure needs," *Integrating Materials and Manufacturing Innovation*, vol. 3, no. 1, p. 12, 2014.

[63] L. Zhang, M. Stratmann, Y. Du, B. Sundman, and I. Steinbach, "Incorporating the CALPHAD sublattice approach of ordering into the phase-field model with finite interface dissipation," *Acta Materialia*, vol. 88, pp. 156–169, 2015.

[64] Z.-K. Liu, "First-principles calculations and CALPHAD modeling of thermodynamics," *Journal of phase equilibria and diffusion*, vol. 30, no. 5, p. 517, 2009.

[65] A. van de Walle and G. Ceder, "Automating first-principles phase diagram calculations," *Journal of Phase Equilibria*, vol. 23, no. 4, p. 348, 2002.

[66] A Van der Ven, J. Thomas, B Puchala, and A. Natarajan, "First-principles statistical mechanics of multicomponent crystals," *Annual Review of Materials Research*, no. 0, 2018.

[67] M Asta, V Ozolins, and C Woodward, "A first-principles approach to modeling alloy phase equilibria," *JOM*, vol. 53, no. 9, pp. 16–19, 2001.

[68] E. S. Kryachko and E. V. Ludeña, "Density functional theory: Foundations reviewed," *Physics Reports*, vol. 544, no. 2, pp. 123–239, 2014.

[69] H. Xing, X. Dong, J. Wang, and K. Jin, "Orientation dependence of columnar dendritic growth with sidebranching behaviors in directional solidification: Insights from phase-field simulations," *Metallurgical and Materials Transactions B*, pp. 1–13, 2018.

[70] T. Takaki, R. Rojas, S. Sakane, M. Ohno, Y. Shibuta, T. Shimokawabe, and T. Aoki, "Phase-field-lattice Boltzmann studies for dendritic growth with natural convection," *Journal of Crystal Growth*, vol. 474, pp. 146–153, 2017.

[71] X. B. Qi, Y. Chen, X. H. Kang, D. Z. Li, and T. Z. Gong, "Modeling of coupled motion and growth interaction of equiaxed dendritic crystals in a binary alloy during solidification," *Scientific Reports*, vol. 7, p. 45 770, 2017.

[72] D. Liu and Y. Wang, "Mesoscale multi-physics simulation of solidification in selective laser melting process using a phase field and thermal lattice boltzmann model," in *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, 2017, V001T02A027–V001T02A027.

[73] A. K. Boukellal, J.-M. Debierre, G. Reinhart, and H. Nguyen-Thi, "Scaling laws governing the growth and interaction of equiaxed al-cu dendrites: A study combining experiments with phase-field simulations," *Materialia*, 2018.

[74] V. Barthelmann, E. Novak, and K. Ritter, "High dimensional polynomial interpolation on sparse grids," *Advances in Computational Mathematics*, vol. 12, no. 4, pp. 273–288, 2000.

[75] E. Novak and K. Ritter, "Simple cubature formulas with high polynomial exactness," *Constructive approximation*, vol. 15, no. 4, pp. 499–522, 1999.

[76] H.-J. Bungartz and M. Griebel, "Sparse grids," *Acta numerica*, vol. 13, pp. 147–269, 2004.

[77] F. Nobile, R. Tempone, and C. G. Webster, "A sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2309–2345, 2008.

[78] G. W. Wasilkowski and H. Woźniakowski, "Explicit cost bounds of algorithms for multivariate tensor product problems," *Journal of Complexity*, vol. 11, no. 1, pp. 1–56, 1995.

[79] C. W. Clenshaw and A. R. Curtis, "A method for numerical integration on an automatic computer," *Numerische Mathematik*, vol. 2, no. 1, pp. 197–205, 1960.

[80] D. Xiu and G. E. Karniadakis, "The Wiener–Askey polynomial chaos for stochastic differential equations," *SIAM Journal on Scientific Computing*, vol. 24, no. 2, pp. 619–644, 2002.

[81] H. N. Najm, "Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics," *Annual Review of Fluid Mechanics*, vol. 41, pp. 35–52, 2009.

[82] R. Askey and J. A. Wilson, *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*. American Mathematical Soc., 1985, vol. 319.

[83] E. Novak and K. Ritter, "High dimensional integration of smooth functions over cubes," *Numerische Mathematik*, vol. 75, no. 1, pp. 79–97, 1996.

[84] D. Xiu and J. S. Hesthaven, "High-order collocation methods for differential equations with random inputs," *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 1118–1139, 2005.

[85] I. Babuška, F. Nobile, and R. Tempone, "A stochastic collocation method for elliptic partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 45, no. 3, pp. 1005–1034, 2007.

[86] F. Nobile, R. Tempone, and C. G. Webster, "An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data," *SIAM Journal on Numerical Analysis*, vol. 46, no. 5, pp. 2411–2442, 2008.

[87] D. Medvedev, F. Varnik, and I. Steinbach, "Simulating mobile dendrites in a flow," *Procedia Computer Science*, vol. 18, pp. 2512–2520, 2013.

[88] S. G. Kim, "A phase-field model with antitrapping current for multicomponent alloys with arbitrary thermodynamic properties," *Acta Materialia*, vol. 55, no. 13, pp. 4391–4399, 2007.

[89] I. Steinbach, "Phase-field models in materials science," *Modelling and simulation in materials science and engineering*, vol. 17, no. 7, p. 073 001, 2009.

[90] O. Shchyglo, E. Borukhovich, P. Engels, M. Stratmann, J. Grler, R. Schiedung, A. Monas, M. Tegeler, R. D. Kamachali, M. K. Rajendran, M. Boeff, D. Medvedev, S. Gladkov, R. Spatschek, and I. Steinbach, *Openphase*, http://www.openphase.de, version 3999, Mar. 13, 2018.

[91] M Stoyanov, "User manual: TASMANIAN sparse grids," Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, Tech. Rep. ORNL/TM-2015/596, 2015.

[92] M. K. Stoyanov and C. G. Webster, "A dynamically adaptive sparse grids method for quasi-optimal interpolation of multidimensional functions," *Computers & Mathematics with Applications*, vol. 71, no. 11, pp. 2449–2465, 2016.

[93] M Stoyanov, "Hierarchy-direction selective approach for locally adaptive sparse grids," Oak Ridge National Laboratory, One Bethel Valley Road, Oak Ridge, TN, Tech. Rep. ORNL/TM-2013/384, 2013.

[94] M. Stoyanov, D. Lebrun-Grandie, J. Burkardt, and D. Munster, *Tasmanian*, Sep. 2013.

[95] S. Suzuki *et al.*, "Topological structural analysis of digitized binary images by border following," *Computer vision, graphics, and image processing*, vol. 30, no. 1, pp. 32–46, 1985.

[96] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[97] A. S. Barros, I. A. Magno, F. A. Souza, C. A. Mota, A. L. Moreira, M. A. Silva, and O. L. Rocha, "Measurements of microhardness during transient horizontal directional solidification of Al-rich Al-Cu alloys: Effect of thermal parameters, primary dendrite arm spacing and Al_2Cu intermetallic phase," *Metals and Materials International*, vol. 21, no. 3, pp. 429–439, 2015.

[98] B. J. Debusschere, H. N. Najm, P. P. Pébay, O. M. Knio, R. G. Ghanem, and O. P. Le Maître, "Numerical challenges in the use of polynomial chaos representations for stochastic processes," *SIAM Journal on Scientific Computing*, vol. 26, no. 2, pp. 698–719, 2004.

[99] B. Debusschere, K. Sargsyan, C. Safta, and K. Chowdhary, "Uncertainty quantification toolkit (UQTk)," *Handbook of Uncertainty Quantification*, pp. 1–21, 2016.

[100] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, "Building skeleton models via 3-D medial surface axis thinning algorithms," *CVGIP: Graphical Models and Image Processing*, vol. 56, no. 6, pp. 462–478, 1994.

[101] J. Mockus, "The Bayesian approach to global optimization," *System Modeling and Optimization*, pp. 473–481, 1982.

[102] ——, "On Bayesian methods for seeking the extremum," in *Optimization Techniques IFIP Technical Conference*, Springer, 1975, pp. 400–404.

[103] D. Huang, T. T. Allen, W. I. Notz, and N. Zeng, "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of Global Optimization*, vol. 34, no. 3, pp. 441–466, 2006.

[104] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv preprint arXiv:0912.3995*, 2009.

[105] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for Gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, 2012.

[106] S. L. Digabel and S. M. Wild, "A taxonomy of constraints in simulation-based optimization," *arXiv preprint arXiv:1505.07881*, 2015.

[107] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints.," in *ICML*, 2014, pp. 937–945.

[108] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," *arXiv preprint arXiv:1403.5607*, 2014.

[109] J. M. Hernández-Lobato, M. Gelbart, M. Hoffman, R. Adams, and Z. Ghahramani, "Predictive entropy search for Bayesian optimization with unknown constraints," in *International Conference on Machine Learning*, 2015, pp. 1699–1707.

[110] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani, "A general framework for constrained bayesian optimization using information-based search," *Journal of Machine Learning Research*, 2016.

[111] S. u. Rehman and M. Langelaar, "Expected improvement based infill sampling for global robust optimization of constrained problems," *Optimization and Engineering*, vol. 18, no. 3, pp. 723–753, 2017.

[112] E. Davis and M. Ierapetritou, "A kriging based method for the solution of mixed-integer nonlinear programs containing black-box functions," *Journal of Global Optimization*, vol. 43, no. 2-3, pp. 191–205, 2009.

[113] J. Müller, C. A. Shoemaker, and R. Piché, "SO-MI: A surrogate model algorithm for computationally expensive nonlinear mixed-integer black-box global optimization problems," *Computers & Operations Research*, vol. 40, no. 5, pp. 1383–1400, 2013.

[114] ——, "SO-I: A surrogate model algorithm for expensive nonlinear integer programming problems including global optimization applications," *Journal of Global Optimization*, vol. 59, no. 4, pp. 865–889, 2014.

[115] J. Müller, "MISO: Mixed-integer surrogate optimization framework," *Optimization and Engineering*, vol. 17, no. 1, pp. 177–203, 2016.

[116] T. Hemker, K. R. Fowler, M. W. Farthing, and O. von Stryk, "A mixed-integer simulation-based optimization approach with surrogate functions in water resources management," *Optimization and Engineering*, vol. 9, no. 4, pp. 341–360, 2008.

[117] B. van Stein, H. Wang, W. Kowalczyk, T. Bäck, and M. Emmerich, "Optimally weighted cluster kriging for big data regression," in *International Symposium on Intelligent Data Analysis*, Springer, 2015, pp. 310–321.

[118] R. B. Gramacy and H. K. H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, vol. 103, no. 483, pp. 1119–1130, 2008.

[119] R. B. Gramacy and H. K. Lee, "Gaussian processes and limiting linear models," *Computational Statistics & Data Analysis*, vol. 53, no. 1, pp. 123–136, 2008.

[120] R. B. Gramacy, M. Taddy, *et al.*, "Categorical inputs, sensitivity analysis, optimization and importance tempering with tgp version 2, an R package for treed Gaussian process models," *Journal of Statistical Software*, vol. 33, no. 6, pp. 1–48, 2010.

[121] C. B. Storlie, H. D. Bondell, B. J. Reich, and H. H. Zhang, "Surface estimation, variable selection, and the nonparametric oracle property," *Statistica Sinica*, vol. 21, no. 2, p. 679, 2011.

[122] Y. Lin and H. H. Zhang, "Component selection and smoothing in smoothing spline analysis of variance models," *Annals of Statistics*, vol. 34, no. 5, pp. 2272–2297, 2006.

[123] ——, "Component selection and smoothing in multivariate nonparametric regression," *The Annals of Statistics*, vol. 34, no. 5, pp. 2272–2297, 2006.

[124] P. Z. G. Qian, H. Wu, and C. J. Wu, "Gaussian process models for computer experiments with qualitative and quantitative factors," *Technometrics*, vol. 50, no. 3, pp. 383–396, 2008.

[125] Q. Zhou, P. Z. Qian, and S. Zhou, "A simple approach to emulation for computer models with qualitative and quantitative factors," *Technometrics*, vol. 53, no. 3, pp. 266–273, 2011.

[126] L. P. Swiler, P. D. Hough, P. Qian, X. Xu, C. Storlie, and H. Lee, "Surrogate models for mixed discrete-continuous variables," in *Constraint Programming and Decision Making*, Springer, 2014, pp. 181–202.

[127] T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree, "Kriging models for global approximation in simulation-based multidisciplinary design optimization," *AIAA journal*, vol. 39, no. 12, pp. 2233–2241, 2001.

[128] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.

[129] J. R. Martins and A. B. Lambe, "Multidisciplinary design optimization: A survey of architectures," *AIAA journal*, vol. 51, no. 9, pp. 2049–2075, 2013.

[130] A. Sóbester, A. I. Forrester, D. J. Toal, E. Tresidder, and S. Tucker, "Engineering design applications of surrogate-assisted optimization techniques," *Optimization and Engineering*, vol. 15, no. 1, pp. 243–265, 2014.

[131] F. A. Viana, T. W. Simpson, V. Balabanov, and V. Toropov, "Special section on multidisciplinary design optimization: Metamodeling in multidisciplinary design

optimization: How far have we really come?" *AIAA Journal*, vol. 52, no. 4, pp. 670–690, 2014.

[132] M. Li, G. Li, and S. Azarm, "A kriging metamodel assisted multi-objective genetic algorithm for design optimization," *Journal of Mechanical Design*, vol. 130, no. 3, p. 031 401, 2008.

[133] H.-L. Jang, H. Cho, K. K. Choi, and S. Cho, "Reliability-based design optimization of fluid–solid interaction problems," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 228, no. 10, pp. 1724–1742, 2014.

[134] Y. Zhang, S. Hu, J. Wu, Y. Zhang, and L. Chen, "Multi-objective optimization of double suction centrifugal pump using kriging metamodels," *Advances in Engineering Software*, vol. 74, pp. 16–26, 2014.

[135] K. Kim, M. Lee, S. Lee, and G. Jang, "Optimal design and experimental verification of fluid dynamic bearings with high load capacity applied to an integrated motor propulsor in unmanned underwater vehicles," *Tribology International*, vol. 114, pp. 221–233, 2017.

[136] Y. Kim, S. Lee, K. Yee, and D.-H. Rhee, "High-to-low initial sample ratio of hierarchical kriging for film hole array optimization," *Journal of Propulsion and Power*, 2017.

[137] J Liu, W.-P. Song, Z.-H. Han, and Y Zhang, "Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models," *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 925–943, 2017.

[138] C. Song, W. Song, and X. Yang, "Gradient-enhanced hierarchical kriging model for aerodynamic design optimization," *Journal of Aerospace Engineering*, vol. 30, no. 6, p. 04 017 072, 2017.

[139] Q. Zhou, Y. Wang, S.-K. Choi, P. Jiang, X. Shao, and J. Hu, "A sequential multi-fidelity metamodeling approach for data regression," *Knowledge-Based Systems*, 2017.

[140] Q. Zhou, Y. Wang, S.-K. Choi, P. Jiang, X. Shao, J. Hu, and L. Shu, "A robust optimization approach based on multi-fidelity metamodel," *Structural and Multidisciplinary Optimization*, pp. 1–23, 2017.

[141] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[142] C. R. Givens, R. M. Shortt, *et al.*, "A class of wasserstein metrics for probability distributions.," *The Michigan Mathematical Journal*, vol. 31, no. 2, pp. 231–240, 1984.

[143] H. B. Nielsen, S. N. Lophaven, and J. Søndergaard, *DACE, a MATLAB Kriging toolbox*. Citeseer, 2002, vol. 2.

[144] N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," *Evolutionary computation*, vol. 11, no. 1, pp. 1–18, 2003.

[145] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and informatics*, vol. 26, pp. 30–45, 1996.

[146] A. H. Gandomi and X.-S. Yang, "Benchmark problems in structural optimization," in *Computational optimization, methods and algorithms*, Springer, 2011, pp. 259–281.

[147] S. S. Rao, *Engineering optimization: theory and practice*. John Wiley & Sons, 2009.

[148] D. Datta and J. R. Figueira, "A real-integer-discrete-coded particle swarm optimization for design problems," *Applied Soft Computing*, vol. 11, no. 4, pp. 3625–3633, 2011.

[149] A Ravindran, G. V. Reklaitis, and K. M. Ragsdell, *Engineering optimization: methods and applications*. John Wiley & Sons, 2006.

[150] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," *Informatica*, vol. 32, no. 3, 2008.

[151] L. R. Meza, S. Das, and J. R. Greer, "Strong, lightweight, and recoverable three-dimensional ceramic nanolattices," *Science*, vol. 345, no. 6202, pp. 1322–1326, 2014.

[152] M. F. Barnsley, *Fractals everywhere*. Academic press, 2014.

[153] A. F. Bower, *Applied mechanics of solids*. CRC press, 2011.

[154] Y. Cho, J.-H. Shin, A. Costa, T. A. Kim, V. Kunin, J. Li, S. Y. Lee, S. Yang, H. N. Han, I.-S. Choi, *et al.*, "Engineering the shape and structure of materials by fractal cut," *Proceedings of the National Academy of Sciences*, vol. 111, no. 49, pp. 17 390–17 395, 2014.

[155] Y. Tang and J. Yin, "Design of cut unit geometry in hierarchical kirigami-based auxetic metamaterials for high stretchability and compressibility," *Extreme Mechanics Letters*, vol. 12, pp. 77–85, 2017.

[156] M. Shahzad, A. Kamran, M. Z. Siddiqui, and M. Farhan, "Mechanical characterization and FE modelling of a hyperelastic material," *Materials Research*, vol. 18, no. 5, pp. 918–924, 2015.

[157] D. Nguyen-Tuong and J. Peters, "Local Gaussian process regression for real-time model-based robot control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, 2008, pp. 380–385.

[158] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local Gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[159] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, "Local Gaussian process regression for real time online model learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 1193–1200.

[160] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Real-time local Gaussian process model learning," in *From Motor Learning to Interaction Learning in Robots*, Springer Berlin Heidelberg, 2010, pp. 193–207.

[161] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, "A survey on metaheuristics for stochastic combinatorial optimization," *Natural Computing*, vol. 8, no. 2, pp. 239–287, 2009.

[162] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced lectures on machine learning*, Springer, 2004, pp. 63–71.

[163] H.-G. Beyer and B. Sendhoff, "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering*, vol. 196, no. 33, pp. 3190–3218, 2007.

[164] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM review*, vol. 53, no. 3, pp. 464–501, 2011.

[165] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," *Journal of Basic Engineering*, vol. 86, no. 1, pp. 97–106, 1964.

[166] A. D. Bull, "Convergence rates of efficient global optimization algorithms," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2879–2904, 2011.

[167] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint arXiv:1012.2599*, 2010.

[168] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[169] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, 2012, pp. 2951–2959.

[170] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," in *Advances in neural information processing systems*, 2014, pp. 918–926.

[171] D. Hernández-Lobato, J. Hernández-Lobato, A. Shah, and R. Adams, "Predictive entropy search for multi-objective Bayesian optimization," in *International Conference on Machine Learning*, 2016, pp. 1492–1501.

[172] P. Hennig and C. J. Schuler, "Entropy search for information-efficient global optimization," *Journal of Machine Learning Research*, vol. 13, no. Jun, pp. 1809–1837, 2012.

[173] Z. Wang, B. Zhou, and S. Jegelka, "Optimization as estimation with gaussian processes in bandit settings," in *Artificial Intelligence and Statistics*, 2016, pp. 1022–1031.

[174] M. Schonlau, W. J. Welch, and D. R. Jones, "Global versus local search in constrained optimization of computer models," *Lecture Notes-Monograph Series*, pp. 11–25, 1998.

[175] J. Parr, A. Keane, A. I. Forrester, and C. Holden, "Infill sampling criteria for surrogate-based optimization with constraint handling," *Engineering Optimization*, vol. 44, no. 10, pp. 1147–1166, 2012.

[176] V. Picheny, R. B. Gramacy, S. Wild, and S. Le Digabel, "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian," in *Advances in Neural Information Processing Systems*, 2016, pp. 1435–1443.

[177] A. Basudhar, C. Dribusch, S. Lacaze, and S. Missoum, "Constrained efficient global optimization with support vector machines," *Structural and Multidisciplinary Optimization*, vol. 46, no. 2, pp. 201–221, 2012.

[178] M. Sacher, R. Duvigneau, O. Le Maître, M. Durand, É. Berrini, F. Hauville, and J.-A. Astolfi, "A classification approach to efficient global optimization in presence of non-computable domains," *Structural and Multidisciplinary Optimization*, pp. 1–21, 2018.

[179] R. B. Gramacy and H. K. H. Lee, "Optimization under unknown constraints," *arXiv preprint arXiv:1004.4027*, 2010.

[180] H Lee, R Gramacy, C. Linkletter, and G Gray, "Optimization subject to hidden constraints via statistical emulation," *Pacific Journal of Optimization*, vol. 7, no. 3, pp. 467–478, 2011.

[181] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[182] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statistics and its Interface*, vol. 2, no. 3, pp. 349–360, 2009.

[183] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[184] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[185] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[186] D. Ginsbourger, R. Le Riche, and L. Carraro, "A multi-points criterion for deterministic parallel global optimization based on Gaussian processes," 2008.

[187] ——, "Kriging is well-suited to parallelize optimization," *Computational Intelligence in Expensive Optimization Problems*, vol. 2, pp. 131–162, 2010.

[188] C. Chevalier and D. Ginsbourger, "Fast computation of the multi-points expected improvement with applications in batch selection," in *International Conference on Learning and Intelligent Optimization*, Springer, 2013, pp. 59–69.

[189] O. Roustant, D. Ginsbourger, and Y. Deville, "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodelling and optimization," *Journal of Statistical Software*, vol. 51, no. 1, 54p, 2012.

[190] S. Marmin, C. Chevalier, and D. Ginsbourger, "Differentiating the multipoint expected improvement for optimal batch design," in *International Workshop on Machine Learning, Optimization and Big Data*, Springer, 2015, pp. 37–48.

[191] ——, "Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors," *arXiv preprint arXiv:1609.02700*, 2016.

[192] B. Letham, B. Karrer, G. Ottoni, and E. Bakshy, "Constrained Bayesian optimization with noisy experiments," *arXiv preprint arXiv:1706.07094*, 2017.

[193] J. Wang, S. C. Clark, E. Liu, and P. I. Frazier, "Parallel Bayesian global optimization of expensive functions," *arXiv preprint arXiv:1602.05149*, 2016.

[194] J. Wu and P. Frazier, "The parallel knowledge gradient method for batch Bayesian optimization," in *Advances in Neural Information Processing Systems*, 2016, pp. 3126–3134.

[195] A. Shah and Z. Ghahramani, "Parallel predictive entropy search for batch global optimization of expensive objective functions," in *Advances in Neural Information Processing Systems*, 2015, pp. 3330–3338.

[196] J. Azimi, A. Fern, and X. Z. Fern, "Batch Bayesian optimization via simulation matching," in *Advances in Neural Information Processing Systems*, 2010, pp. 109–117.

[197] J. Azimi, A. Fern, X. Zhang-Fern, G. Borradaile, and B. Heeringa, "Batch active learning via coordinated matching," *arXiv preprint arXiv:1206.6458*, 2012.

[198] J. Azimi, A. Jalali, and X. Fern, "Hybrid batch Bayesian optimization," *arXiv preprint arXiv:1202.5597*, 2012.

[199] T. Desautels, A. Krause, and J. W. Burdick, "Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3873–3923, 2014.

[200] E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis, "Parallel Gaussian process optimization with upper confidence bound and pure exploration," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2013, pp. 225–240.

[201] J. González, Z. Dai, P. Hennig, and N. Lawrence, "Batch Bayesian optimization via local penalization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 2016, pp. 648–657.

[202] T. Kathuria, A. Deshpande, and P. Kohli, "Batched Gaussian process bandit optimization via determinantal point processes," in *Advances in Neural Information Processing Systems*, 2016, pp. 4206–4214.

[203] Z. Wang, C. Li, S. Jegelka, and P. Kohli, "Batched high-dimensional Bayesian optimization via structural kernel learning," *arXiv preprint arXiv:1703.01973*, 2017.

[204] N. Rontsis, M. A. Osborne, and P. J. Goulart, "Distributionally robust optimization techniques in batch Bayesian optimization," *arXiv preprint arXiv:1707.04191*, 2017.

[205] V. Nguyen, S. Rana, S. K. Gupta, C. Li, and S. Venkatesh, "Budgeted batch Bayesian optimization," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, IEEE, 2016, pp. 1107–1112.

[206] E. A. Daxberger and B. K. H. Low, "Distributed batch Gaussian process optimization," in *International Conference on Machine Learning*, 2017, pp. 951–960.

[207] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, no. 7, 2008.

[208] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *International Conference on Parallel Problem Solving from Nature*, Springer, 2004, pp. 282–291.

[209] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," in *International Conference on Parallel Problem Solving from Nature*, Springer, 1994, pp. 249–257.

[210] K. V. Pagalthivarthi, J. M. Furlan, and R. J. Visintainer, "Wear rate prediction in multi-size particulate flow through impellers," in *ASME 2013 Fluids Engineering Division Summer Meeting*, American Society of Mechanical Engineers, 2013.

[211] K. Pagalthivarthi and G. Addie, "Prediction methodology for two-phase flow and erosion wear in slurry impellers," in *4th International Conference on Multiphase Flow*, 2001.

[212] P. R. Spalart, S. R. Allmaras, *et al.*, "A one equation turbulence model for aerodynamic flows," *Recherche Aerospatiale-French Edition*, pp. 5–5, 1994.

[213] K. Pagalthivarthi and R. Visintainer, "Finite element prediction of multi-size particulate flow through three-dimensional channel: Code validation," *The Journal of Computational Multiphase Flows*, vol. 5, no. 1, pp. 57–72, 2013.

[214] F. White, *Viscous Fluid Flow 2nd Edition*. McGraw-Hill New York, 1991.

[215] T. J. Hughes and A. N. Brooks, "A theoretical framework for petrov-galerkin methods with discontinuous weighting functions: Application to the streamline-upwind procedure," *Finite elements in fluids*, pp. 47–65, 1982.

[216] O. Schenk and K. Gärtner, "Solving unsymmetric sparse systems of linear equations with PARDISO," *Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.

[217] H. H. Tian, G. R. Addie, and K. V. Pagalthivarthi, "Determination of wear coefficients for erosive wear prediction through coriolis wear testing," *Wear*, vol. 259, no. 1-6, pp. 160–170, 2005.

[218] S. N. Lophaven, H. B. Nielsen, and J. Søndergaard, *Aspects of the matlab toolbox DACE*. Citeseer, 2002.

[219] B. Gauthier and L. Pronzato, "Convex relaxation for IMSE optimal design in random-field models," *Computational Statistics & Data Analysis*, vol. 113, pp. 375–394, 2017.

[220] R. T. Silvestrini, D. C. Montgomery, and B. Jones, "Comparing computer experiments for the Gaussian process model using integrated prediction variance," *Quality Engineering*, vol. 25, no. 2, pp. 164–174, 2013.

[221] B. Gauthier and L. Pronzato, "Spectral approximation of the IMSE criterion for optimal designs in kernel-based interpolation models," *SIAM/ASA Journal on Uncertainty Quantification*, vol. 2, no. 1, pp. 805–825, 2014.

[222] B. Ankenman, B. L. Nelson, and J. Staum, "Stochastic kriging for simulation metamodeling," *Operations research*, vol. 58, no. 2, pp. 371–382, 2010.

[223] A. Tran and Y. Wang, *RMD: Reliable molecular dynamics*, `https://github.com/GeorgiaTechMSSE/ReliableMD`, 2017.

[224] ——, "Reliable Molecular Dynamics: Uncertainty quantification using interval analysis in molecular dynamics simulation," *Computational Materials Science*, vol. 127, pp. 141–160, 2017.

[225] ——, "Quantifying model-form uncertainty in molecular dynamics simulation," *TMS 2016 Supplemental Proceedings*, p. 283, 2016.

[226] ——, "Molecular dynamics simulation with interval-valued interatomic potentials," in *Proceedings of the ASME 2016 International Design Engineering Technical Conference & Computers and Information in Engineering Conference IDETC/CIE*, American Society of Mechanical Engineers, 2016.

VITA

Anh Vuong Tran obtained his Bachelor of Science in Mechanical Engineering at Georgia Institute of Technology in 2011, Master of Science in Applied Mathematics at Georgia Southern University in 2014, and Doctor of Philosophy in Mechanical Engineering at Georgia Institute of Technology in 2018. His research interests include uncertainty quantification and optimization under uncertainty, machine learning (shallow/deep learning), and applications in computational solid/fluid mechanics.